COPYRIGHT AND CITATION CONSIDERATIONS FOR THIS THESIS/DISSERTATION

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

- NonCommercial — You may not use the material for commercial purposes.

- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

How to cite this thesis
Worldwide Interoperability for Microwave Access performance simulations using Markov model and simulators

By

Sandla Nkaule
820400468

A dissertation submitted in partial fulfilment of the degree:

M Tech Electrical Engineering

in the

Department of Electrical and Electronic Engineering Technology

at the

Faculty of Engineering and the Built Environment

University of Johannesburg

Supervisor: Prof. Meera K. Joseph

June 2017

Johannesburg, South Africa
Declaration

I Sandla Nkaule hereby declare that the dissertation titled “Worldwide Interoperability for Microwave Access performance simulations using Markov model and simulators” submitted for MTech Electrical and Electronic Engineering Technology at the University of Johannesburg is my original work. This dissertation has not been previously submitted to another University or Higher Institution for a degree.

Date:....02 November 2017.....
Acknowledgement

I am humbled by the grace that our Lord Jesus Christ has extended and His mercy for life, health and strength to pursue MTech.

I would like to extend my acknowledgement to UJ / CSIR for funding my studies in MTech and also would like to thank FEBE and FHDC for giving me this research opportunity. I thank my wonderful supervisor Prof. Meera K. Joseph who possess immeasurable patience, for her undivided attention, support and guidance in my studies.

My family, the Nkaules, without mentioning individuals, are my pillars and support system and I am grateful to God for choosing them as my family. And lastly my children whom God has gifted me with to expand my sense of purpose.

List of Publications

Abstract

Wireless communication has been part of communication for decades and continues to evolve in a radical manner. These days communication cannot be separated with various technologies and newly developed devices tend to be integrated to support wireless. WiMAX is a wireless industry coalition and is part of the latest wireless infrastructure which includes 3G and LTE and needs to supports both communication and other technologies. WiMAX is seen as WiFi that supports greater distances of up to about 50 kms in radius. In this study we simulate the WiMAX performance - simulations are done using OPNET, NS-3 and Markov Chains are also used to evaluate the performance and compare the results. The study looks at the delay and the throughput in the WiMAX network as we increase numbers of connected devices from 20 to 50, 75 and lastly 100 mobile devices. The results of these performance indicators were analyzed and OPNET was found to be the best simulation tool amongst the three that was user friendly and easy to work with, which also returned results that are satisfactory and resembles reality.

Keywords: WiMAX performance, IEEE 802.16, Markov model, OPNET, NS-3, simulation.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>3G</td>
<td>Third Generation</td>
</tr>
<tr>
<td>4G</td>
<td>Fourth Generation</td>
</tr>
<tr>
<td>AAA</td>
<td>Authentication, Authorization and Accounting</td>
</tr>
<tr>
<td>ACK</td>
<td>Positive Acknowledgement</td>
</tr>
<tr>
<td>AMC</td>
<td>Automatic Modulation Classifier</td>
</tr>
<tr>
<td>ASN</td>
<td>Access Service Network</td>
</tr>
<tr>
<td>ASP</td>
<td>Application Service Provider</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BCH</td>
<td>Bose-Chaudhuri-Hocquenqhen</td>
</tr>
<tr>
<td>BER</td>
<td>Bit-Error-Rate</td>
</tr>
<tr>
<td>BMAP</td>
<td>Batch Markov Arrival Process</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CC</td>
<td>Convolutional Code</td>
</tr>
<tr>
<td>CLI</td>
<td>Command Line Interface</td>
</tr>
<tr>
<td>CPS</td>
<td>Common Part Sub-layer</td>
</tr>
<tr>
<td>CRC</td>
<td>Cyclic Redundancy Check</td>
</tr>
<tr>
<td>CS</td>
<td>Convergence Sub-layer</td>
</tr>
<tr>
<td>CSN</td>
<td>Connectivity Service Network</td>
</tr>
<tr>
<td>CTMC</td>
<td>Continuous Time Markov Chain</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>DTMC</td>
<td>Discrete Time Markov Chain</td>
</tr>
<tr>
<td>DVB-H</td>
<td>Digital Video Broadcast-Handheld</td>
</tr>
<tr>
<td>FDD</td>
<td>Frequency Division Duplexing</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transforms</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Machines</td>
</tr>
<tr>
<td>GHz</td>
<td>Giga Hertz</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>HA</td>
<td>Home Agent</td>
</tr>
<tr>
<td>H-NSP</td>
<td>Home Network Service Provider</td>
</tr>
<tr>
<td>H-NSP</td>
<td>Home-NSP</td>
</tr>
<tr>
<td>HSS</td>
<td>Home subscriber server</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineer</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMT</td>
<td>International Mobile Telecommunications</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medical band</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Unit</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LoS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
<tr>
<td>MATLAB</td>
<td>Matrix Laboratory</td>
</tr>
<tr>
<td>MDP</td>
<td>Markov decision process</td>
</tr>
<tr>
<td>MMR</td>
<td>Mobile Multihop Relay</td>
</tr>
<tr>
<td>MS</td>
<td>Mobile Stations</td>
</tr>
<tr>
<td>NACK</td>
<td>Negative Acknowledgement</td>
</tr>
</tbody>
</table>
NAP  Network Access Provider
NLOS  Non Line of Sight Propagation
NPS  Number of Polls
NRM  Network Reference Model
NS-3  Network Simulator 3
NSP  Network Service Providers
OFDM  Orthogonal Frequency Division Multiplexing
OFDMA  Orthogonal Frequency Division Multiple Accesses
OPNET  Optimized Network Engineering Tool
OSI  The Open Systems Interconnection model
PDU  Protocol Data Unit
PHY  Physical Standard
PKM  Privacy Key Management
PMP  Point to Multipoint
QAM  Quadrature Amplitude Modulation
QoS  Quality of Service
QPSK  Quadrature Phase Shift Keying
RADIUS  Remote Authentication Dial-in User Service
RP  Reference Points
RS  Reed-Solomon
SAP  Service Access Points
SC  Single Carrier
SDU  Service Data Units
SN  Subscriber Node
SNR  Signal-to-Noise Ratio
SOFDMA  Scalable Orthogonal Frequency Division Multiplexing Access
SS  Subscriber Station
SSN  Subscriber Sub Node
TDD  Time Division Duplex
UGS  Unsolicited Grant Service
VLAN  Virtual Local Area Network
V-NPS  Visited NSP
WFA  Wi-Fi Alliance
WiFi  Wireless Fidelity
WiMAX  Worldwide Interoperability for Microwave Access
WMAN  Wireless Metropolitan Network
List of Figures

Figure 1: WiMAX Network Infrastructure .............................................................................. 5
Figure 2: PHY and MAC layers of WiMAX specification showing MAC privacy sublayer ....... 7
Figure 3: WiMAX Deployments ............................................................................................. 9
Figure 4: Network Reference Model (NRM)............................................................................ 9
Figure 5: WiMAX Protocol Layers ....................................................................................... 12
Figure 6: OFDM transmitter .................................................................................................. 14
Figure 7: OFDM receiver ....................................................................................................... 14
Figure 8: Frequency domain representation of OFDM symbol .............................................. 15
Figure 9: Example of a data region that defines an OFDMA allocation ................................ 15
Figure 10: Markov Property.................................................................................................... 19
Figure 11: Gilbert's Channel Model ...................................................................................... 20
Figure 12: The Markov state diagram of the Gilbert-Elliott model ....................................... 21
Figure 13: Single-error state model and its transition probability matrix .............................. 21
Figure 14: Probability transition diagram for 2-state Markov chain ................................... 24
Figure 15: Probability transition diagram for 3-state Markov chain .................................. 24
Figure 16: Probability transition diagram for 4-state Markov chain .................................. 24
Figure 17: 8 Steps of Simulation ......................................................................................... 33
Figure 18: Windows Start Button and All Programs ............................................................ 37
Figure 19: List of All Programs installed in the ..................................................................... 37
Figure 20: OPNET Modeler 14.5 sub programs ................................................................... 38
Figure 21: OPNET Modeler 14.5 Startup Window ............................................................... 38
Figure 22: OPNET Modeler 14.5 Main Window .................................................................. 39
Figure 23: Starting a project and giving a project a name .................................................. 40
Figure 24: Startup Wizard Initial Topology Window ............................................................. 40
Figure 25: OPNET Modeler 14.5 Initial Window ................................................................. 41
Figure 26: Topology Selection Window ................................................................................. 41
Figure 27: Wireless Network Deployment Window ............................................................. 42
Figure 28: Wireless Deployment Wizard ............................................................................. 42
Figure 29: Configuring Location .......................................................................................... 43
Figure 30: Technology Selection .......................................................................................... 43
Figure 31: Technology Specification Windows and selection of Pathloss ............................ 44
Figure 32: Topology Selection Windows .............................................................................. 44
Figure 33: Topology Configuration ........................................................................................ 45
Figure 34: Node Mobility Window ....................................................................................... 45
Figure 35: Configuration Summary Window ......................................................................... 46
Figure 36: Configured WiMAX Network Model ................................................................. 46
Figure 37: Selecting Application Server .............................................................................. 47
Figure 38: Selecting a link for Application Server to the Backbone ..................................... 47
Figure 39: Adding link between backbone and Application Server ..................................... 48
Figure 40: WiMAX Network Model for 20 Subscriber Stations ......................................... 48
Figure 125: Running simulation in terminal and Python after pressing the Simulate button.... 106
Figure 126: Simulation running in terminal and Python. .......................................................... 107
Figure 127: Simulation of 75 SS progressing in terminal. .......................................................... 107
Figure 128: Simulation of 75SS progressing in Python. ............................................................ 108
Figure 129: Loading plot files for 75 SS. ................................................................................... 109
Figure 130: Delay of the WiMAX System at 75 SS................................................................. 110
Figure 131: Throughput of the WiMAX System at 75 SS....................................................... 111
Figure 132: NetAnim output with packets flow from Applications Server to BS................. 111
Figure 133: NetAnim output with packets from BS to 75 Subscriber Stations. .................... 112
Figure 134: Running simulation using the ./waf for 100 SS...................................................... 113
Figure 135: Terminal and Python output before pressing Simulate button. ......................... 113
Figure 136: Terminal and Python output after pressing Simulate button............................. 114
Figure 137: Python output after pressing Simulate button. .................................................... 114
Figure 138: Terminal output after pressing Simulate button. ............................................... 115
Figure 139: gnuplot command and gnuplot workspace, loading .plt files. ......................... 115
Figure 140: Delay for the WiMAX System at 100 SS............................................................ 116
Figure 141: Throughput of the WiMAX System at 100 SS................................................... 117
Figure 142: NetAnim simulation, WiMAX traffic from BS to SS nodes............................. 117
Figure 143: NetAnim simulation, WiMAX traffic from Applications Server to BS ............ 117
Figure 144: WiMAX Network Model ..................................................................................... 31

List of Tables
Table 1: The family of the IEEE 802.16 Standards................................................................. 8
Table 2: Parameters for the Subscriber Stations................................................................. 34
Table 3: Parameters for the Base Stations ............................................................................ 34
Table 4: Simulation Results - Delay......................................................................................... 118
Table 5: Simulation Results - Throughput ............................................................................ 118
<table>
<thead>
<tr>
<th>Equation</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>78</td>
</tr>
<tr>
<td>11</td>
<td>78</td>
</tr>
<tr>
<td>12</td>
<td>78</td>
</tr>
<tr>
<td>13</td>
<td>78</td>
</tr>
<tr>
<td>14</td>
<td>78</td>
</tr>
<tr>
<td>15</td>
<td>78</td>
</tr>
<tr>
<td>16</td>
<td>79</td>
</tr>
<tr>
<td>17</td>
<td>79</td>
</tr>
<tr>
<td>18</td>
<td>79</td>
</tr>
<tr>
<td>19</td>
<td>79</td>
</tr>
<tr>
<td>20</td>
<td>79</td>
</tr>
<tr>
<td>21</td>
<td>79</td>
</tr>
<tr>
<td>22</td>
<td>80</td>
</tr>
<tr>
<td>23</td>
<td>80</td>
</tr>
<tr>
<td>24</td>
<td>80</td>
</tr>
</tbody>
</table>
# Table of Contents

1 INTRODUCTION .......................................................................................................................... 1
   1.1 Background Review ............................................................................................................. 1
   1.2 Aim of the Research ........................................................................................................... 1
   1.3 Problem Statement ............................................................................................................. 1
   1.4 Research Question ............................................................................................................. 2
   1.5 Research Objective .............................................................................................................. 2
   1.6 Overview .............................................................................................................................. 2
   1.7 Limitations of the Study .................................................................................................... 3
   1.8 Research Methodology and Design .................................................................................. 3
   1.9 Ethical Considerations ...................................................................................................... 3
   1.10 Significance and Unique Contributions ............................................................................. 3
   1.11 Conclusion ....................................................................................................................... 4

2 LITERATURE REVIEW ............................................................................................................... 5
   2.1 Introduction ......................................................................................................................... 5
   2.2 WiMAX .................................................................................................................................. 6
      2.2.1 Brief history of WiMAX .............................................................................................. 6
      2.2.2 IEEE 802.16 standard and WiMAX .......................................................................... 6
      2.2.3 WiMAX Architecture ............................................................................................... 8
      2.2.4 WiMAX Protocol Layers .......................................................................................... 12
      2.2.5 WiMAX Performance ............................................................................................... 15
      2.2.6 Quality of Service in WiMAX Networks .................................................................... 16
   2.3 Markov Chains ..................................................................................................................... 18
      2.3.1 History of Markov Chains .......................................................................................... 18
      2.3.2 Markov Model ............................................................................................................ 18
      2.3.3 Markov Property ....................................................................................................... 18
      2.3.4 Generative Channel Models ..................................................................................... 18
      2.3.5 Continuous Time Markov Chains .......................................................................... 19
      2.3.6 Discrete Time Markov Chains (DTMC) .................................................................... 22
      2.3.7 Elements of Markov chains ...................................................................................... 22
   2.4 Simulation .............................................................................................................................. 25
      2.4.1 Network Simulator-3 (NS-3) ..................................................................................... 25
      2.4.2 OPNET ....................................................................................................................... 26
   2.5 Similar Work ........................................................................................................................ 28
      2.5.1 Simulation of WiMAX with NS-3 ............................................................................. 28
      2.5.2 Simulation of WiMAX with OPNET ......................................................................... 28
      2.5.3 Modelling WiMAX PHY Layer with Markov Chain .............................................. 29

3 RESEARCH METHODOLOGY ................................................................................................. 30
   3.1 Introduction ........................................................................................................................ 30
   3.2 Approach and Design ......................................................................................................... 30
      3.2.1 Topology .................................................................................................................... 31
      3.2.2 Protocols and Applications ....................................................................................... 31
      3.2.3 Expected Simulation Outcome .................................................................................. 31
      3.2.4 Simulation Parameters .............................................................................................. 31
3.3 Simulation with NS-3 ................................................................. 32
  3.3.1 NS-3 Simulation Reliability ............................................... 33
3.4 Simulation with OPNET ................................................................ 34
  3.4.1 Accuracy of OPNET Modeler ............................................. 35
3.5 Markov Modelling of WiMAX .................................................... 35

4 SIMULATION USING OPNET MODELER 14.5 .......................... 36
  4.1 Introduction ............................................................................ 36
  4.2 Starting the OPNET Modeler 14.5 Simulation Program .......... 36
  4.3 Simulation for Twenty (20) Subscriber Stations (SS) ............. 39
    4.3.1 Creating Network Model ............................................... 39
    4.3.2 Choosing Statistics ....................................................... 49
    4.3.3 Running the simulation ............................................... 54
    4.3.4 View and analyse results ............................................. 56
  4.4 Simulation for Fifty (50) Subscriber Stations (SS) ................. 60
    4.4.1 Creating Network Model ............................................... 60
    4.4.2 Choosing Statistics and Running the simulation ............. 61
    4.4.3 View and analyse results ............................................. 61
  4.5 Simulation for Seventy Five (75) Subscriber Stations (SS) ....... 64
    4.5.1 Creating Network Model ............................................... 64
    4.5.2 View and analyse results ............................................. 65
  4.6 Simulation for Hundred (100) Subscriber Stations (SS) .......... 68
    4.6.1 Creating Network Model ............................................... 68
    4.6.2 View and analyse results ............................................. 69

5 USE OF MARKOV MODEL TO ANALYSE PERFORMANCE ....... 73
  5.1 Introduction ............................................................................ 73
  5.2 Similar Work: Markov Models for Network Simulations ......... 73
  5.3 Modelling of WiMAX with Markov Chain ............................. 74
    5.3.1 Performance Analysis for WiMAX .................................. 75
  5.4 Simulation with MATLAB R2011a ............................................ 78
    5.4.1 MATLAB .................................................................... 78
    5.4.2 Throughput Simulation .................................................. 79
    5.4.3 Delay equation in MATLAB ......................................... 81
    5.4.4 Conclusion .................................................................... 83

6 SIMULATION USING NS-3.19 ...................................................... 84
  6.1 Introduction ............................................................................ 84
  6.2 Simulation of Twenty Subscriber Stations .............................. 85
    6.2.1 Converting the nodes of a network model to a C++ program ........ 85
    6.2.2 Running program in Command Line Interface (CLI) using waf command of NS-3 87
    6.2.3 Run Simulation .............................................................. 90
    6.2.4 Plotting Graph of the simulation output in CLI Gnuplot ....... 92
  6.3 Simulation of Fifty Subscriber Stations .................................... 98
    6.3.1 Converting the nodes of a network model to a C++ program ........ 99
    6.3.2 Running program in Command Line Interface (CLI) using waf command of NS-3 100
6.3.3 Run Simulation ..................................................................................................... 100
6.3.4 Plotting Graph of the simulation output in CLI Gnuplot ....................................... 101
6.3.5 Animate using NetAnim ....................................................................................... 103
6.4 Simulation of Seventy Five Subscriber Stations......................................................... 104
6.4.1 Converting the nodes of a network model to a C++ program ............................... 104
6.4.2 Running program in Command Line Interface (CLI) using waf command of NS-3 105
6.4.3 Run Simulation ..................................................................................................... 105
6.4.4 Plotting Graph of the simulation output in CLI Gnuplot ....................................... 107
6.4.5 Animation using NetAnim .................................................................................... 110
6.5 Simulation of Hundred Subscriber Stations ................................................................ 111
6.5.1 Converting the nodes of a network model to a C++ program ............................... 112
6.5.2 Running program in Command Line Interface (CLI) using waf command of NS-3 112
6.5.3 Run Simulation ..................................................................................................... 112
6.5.4 Plotting Graph of the simulation output in CLI Gnuplot ....................................... 114
6.5.5 Animation using NetAnim .................................................................................... 116
7 DISCUSSION OF RESULTS ............................................................................................. 118
7.1 Comparison of Results ................................................................................................. 118
7.2 Comparison with previous studies .................................................................................. 118
7.2.1 Simulation of WiMAX System Performance using OPNET ................................. 118
7.2.2 Simulation of WiMAX System Performance using NS-3 ...................................... 119
7.2.3 Simulation of WiMAX System Performance using Markov Chain ..................... 119
8 CONCLUSION AND FUTURE WORK ........................................................................... 120
8.1 Conclusion ....................................................................................................................... 120
8.2 Future Work ..................................................................................................................... 120
9 REFERENCES ................................................................................................................... 121
1 INTRODUCTION
1.1 Background Review
The communication infrastructure worldwide including underdeveloped countries has evolved from wired to wireless and that gave rise to Radio Links, Wireless Local Area Networks (WLAN), Wireless Fidelity (WiFi), Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX) [103] [7] [31]. The implementation of these has improved the way society communicates in terms of speed of communication between a sender and a receiver, and the cost of communication has also significantly dropped and that allows the vast rural community to be part of these technologies.

WiMAX infrastructure has been in use in some parts of the world especially in South Korea and in South Africa it is provided by Telkom and Neotel. The availability of WiMAX can significantly reduce cost of electronic communication for the rural community as this is wireless and required less cabling and digging. Information sharing cost for the rural communities can be reduced to zero as they will be compared to a local area network which costs nothing to communicate. Electronic communication and social technologies can be accessible to the poorest communities as the cost of these can be significantly reduced by using WiMAX. One of the reasons WiMAX is becoming popular in the wireless community is that it offers high through-put broadband connection over a long distance, but it does not come without challenges though. The performance of WiMAX is a major concern that affects the throughput and the delay of infrastructure.

1.2 Aim of the Research
The aim of the research is to analyse the performance of the WiMAX during the transmission of the following data types:

- Video
- Voice
- HTTP

Simulation software that will be used in this study is the following:

- OPNET
- NS-3
- MATLAB (Markov Chains)

1.3 Problem Statement
As stated in the previous sections of this chapter, in this research WiMAX performance will be analysed and simulated. In our study we will simulate a WiMAX network that connects to Internet for communicating with an application server that provides i) Video, ii) Voice, iii) and HTTP traffic. The study will simulate four (4) scenarios. The scenarios are as follows:

- Scenario 1 – WiMAX Network with 20 workstations
- Scenario 2 – WiMAX Network with 50 workstations
- Scenario 3 – WiMAX Network with 75 workstations
• Scenario 4 – WiMAX Network with 100 workstations

In these scenarios we will mainly simulate the delay and the throughput of the WiMAX network using simulators (OPNET, NS-3) and Markov chains (using MATLAB). The main problem of the system is performance and the study will analyze the system performance as the number of nodes are increased

1.4 Research Question
How do we simulate and analyze the performance of WiMAX using various simulators and Markov modelling?

1.5 Research Objective
The main objective of this research is to understand the performance of WiMAX through simulations. The following is the sub-objectives:

- To simulate the WiMAX performance using:
  - Markov Model - MATLAB
  - Simulators – OPNET and NS-3
- To compare the WiMAX performance amongst the above mentioned simulators.
- To measure performance evaluation with existing work.

1.6 Overview
This section outlines the structure of the thesis and how chapters are organised:

- **Chapter 1**: This chapter introduces the study related to WiMAX, giving an insight to aim and the objective of the study. We also provides a methodology and design, limitations of the study and ethical considerations to the unique contribution and overview of the thesis.
- **Chapter 2**: This chapter introduces the background knowledge and history of the 802.16 standard and WiMAX, also the Simulators and Markov Chains are introduced and reviewed.
- **Chapter 3**: This chapter discusses the methodology used in this study, including simulators and the network model that will be simulated.
- **Chapter 4**: In this chapter we install the OPNET software and run it to simulate the WiMAX network.
- **Chapter 5**: Simulation using Markov Chains and retransmission probability, we also introduce the MATLAB software and how to create a script from Markov mathematical results and running them in MATLAB.
- **Chapter 6**: In this chapter we simulate the WiMAX network with NS-3, installing the software in Linux, explaining steps to follow on how to simulate with the software also creating scripts that simulate WiMAX Network.
- **Chapter 7**: Discussion of simulated results.
- **Chapter 8**: This is the conclusion chapter and here we reflect on the simulations and comparisons also make recommendation for further study.
1.7 Limitations of the Study
The network topology for the WiMAX is limited to 100 nodes per base station for OPNET software. We will therefore limit the simulation nodes 100. The nodes represent a number of computers or the number of links that connect to the base station. We will first simulate the network using 20 nodes, then 50, then 75 and lastly 100 nodes.
The simulations are limited to Video, Voice and HTTP data as these are the one of the available options in the simulation software and these are the popular types of traffic in networking. For all simulators, the performance parameters that we will seek to analyse are the throughput and the delay in the WiMAX network as we increase the number of nodes.

1.8 Research Methodology and Design
In this research we simulate the performance of WiMAX using various simulators and the steps are further illustrated in Figure 17. In this study we will focus on the transmission of data within the WiMAX medium from the source of data to the destination. The simulation will be done by Markov Chains, NS-3 and OPNET, then we do a comparison of results from both simulators.
The research method to be employed is experimental research, this research method aids in controlling independent variables for the experiments aim to remove extraneous and unwanted variables. Due to the control set up by experimenter and the strict conditions, better results can be achieved. Another good thing about experimental research is that experiments can be repeated and results can be checked again. Better results that have been obtained can also give researcher greater confidence regarding the results [65].
Based on simulations, the OPNET graph would display an output such that the Y-axis becomes the simulated variable and the X-axis becomes the time, we will therefore simulate four scenarios in OPNET. In NS-3 simulation software the Y-axis simulated variable and the X-axis displays a number of nodes, this is not the same the situation as the OPNET. In MATLAB (using the Markov Chain) we will have the X-axis as throughput or delay then X-axis as the number of nodes. The calculation in the Markov Chain is different from the other two simulators and these will be further explained as we get into details of the simulation using Markov Chain.

1.9 Ethical Considerations
During the period of the study we have considered the following ethical guidelines:
- Information obtained from external source will be fairly used and cited.
- Video, audio and http data used will be generated by the simulators

1.10 Significance and Unique Contributions
Similar simulation of the performance of WiMAX using various simulators and Markov modelling was not done before using same parameters. The study is about the performance of the WiMAX network, this can assist in feasibility of implementation of the infrastructure especially in rural areas of the South African region where there is no 3G coverage.
The performance study of the WiMAX network will assist in understanding the number of users and subscriber stations that can be connected to a single WiMAX base station. Also how the base station will behave when these subscribers are increased, this will assist in analyzing the handling capacity of the base station.

1.11 Conclusion
At first we introduced the study by providing the background outlining the importance of communication and how it evolved in terms of technological infrastructure and devices used for communication. We then progressed to introduce WiMAX as one of the communication channels and how beneficial it is to the society. We have also touched the WiMAX infrastructure that we will use during the study and also type of simulators that we will use to conduct our study. The type of traffic was also discussed and the parameters that we will look at when we measure the performance of the WiMAX network. Lastly we outlined the ethical guidelines that we will follow and the significance and the uniqueness of this study.
2 LITERATURE REVIEW

2.1 Introduction

The permanent evolution of future wireless network technologies together with demand for new multimedia applications, has driven a need to create new wireless, mobile and multimedia-awareness systems [13]. WiMAX [103] technology was created by the WiMAX forum in June 2001 to promote conformity and interoperability of the standard. The forum describes WiMAX as a standards based technology enabling last mile delivery of wireless broadband access as an alternative to cable and DSL [8].

The other standards addresses the transmissions over a single frequency range, while WiMAX allows data transport over multiple broad frequency ranges. This maximizes the technology’s ability to transmit over the frequencies that will avoid interfering with other wireless applications [1].

As seen in Figure 1, the WiMAX network infrastructure promises to connect a wide range of scenarios to the Internet and can enable them to have a swift wireless communication as in a case of WiFi, so we can conveniently see a WiMAX as a WiFi in steroids.

As this research is based on the simulation of the WiMAX and modelling with Markov Chains, we also elaborate on the following topics:

- WiMAX
- Markov
- Simulators: OPNET, Markov Chains (using MATLAB) and NS-3
2.2 WiMAX

2.2.1 Brief history of WiMAX

As the evolution of wireless technologies continues toward realization of mobile broadband access to content-rich multimedia services, the communication industry is going through a significant change. The emphasis is to have a communication infrastructure that is low cost and simple, also the desire is to have an architecture that supports internet-based protocols and applications [2].

In 1990, the Institute of Electrical and Electronics Engineer (IEEE), formed a new working group, 802.11 with focus on indoor wireless to operate in the existing unlicensed bands. The devices that operate using the standard were manufactured in the middle 90s, and in 1999 the “IEEE 802.11b” was created. Wireless Fidelity (WiFi, known as WFA today) was created in an efforts to guarantee interoperability between different implementations, regardless of the manufacturer [3].

A new effort in the wireless arena is taken place known as IEEE 802.16 (WiMAX). In 2004, the original standard, IEEE 802.16 (2001), was extended to operate in a large range of frequencies (2 to 11 GHz) range even coexisting with WiFi. WiMAX aims to provide the ultimate solution for broadband outdoor wireless, it is targeted to large operators in rural and semi urban setups, are far from low cost. Although the technology can operate in a wide range of frequencies, it is still subject to regulatory restrictions [3].

2.2.2 IEEE 802.16 standard and WiMAX

WiMAX which is often called Wireless Metropolitan Network (WMAN) was adopted by WiMAX Forum formed in 2001, The WiMAX forum’s primary goal is to accelerate the adoption, deployment and expansion of WiMAX technologies across the globe while facilitating roaming agreements, sharing best practices [4].

The IEEE 802.16 group addresses wireless technology application to link commercial and residential buildings to high-rate core networks and thereby provide access to those networks and the link was called “last mile” [5], IEEE 802.16 is a unit of the IEEE 802 LAN/MAN Standards Committee and provides a single carrier (SC) physical standard (PHY). From The IEEE 802.16 group, the 802.16a was developed, and this standard included the NLOS applications in the 2GHz–11GHz band, using an Orthogonal Frequency Division Multiplexing (OFDM)-based physical layer. Older versions of this were then replaced by the IEEE 802.16-2004 which formed the basis for the first WiMAX solution. These early WiMAX solutions based on IEEE 802.16-2004 are referred to as fixed WiMAX [16].
Various physical layers were combined with the original IEEE 802.16 standard medium access control protocols, however, also changes in the data link layer protocols were made [19]. Table 1 provides some useful information on the evolution of the IEEE 802.16 standards, for more information related to the amendments and revisions of the IEEE 802.16 standard.
Table 1: The family of the IEEE 802.16 Standards. [7]

<table>
<thead>
<tr>
<th>Year</th>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>802.16k-2007 [387]</td>
<td>IEEE Standard for Local and Metropolitan Area Networks Media Access Control (MAC) Bridges - Amendment 5: Bridging of IEEE 802.16</td>
</tr>
</tbody>
</table>

2.2.3 WiMAX Architecture

WiMAX architectures use either Point to Multipoint (PMP), Mesh, or Mobile Multihop Relay (MMR) mode, MMR shown in Figure 3 adopted from [12]. In PMP subscriber stations connects to the base station in a single hop route. And in Mesh mode subscriber stations can communicate in an ad-hoc fashion. The Mobile Multi-hop Relay (MMR) mode was introduced as an extension for PMP mode as in Figure 3 [12].
2.2.3.1 WiMAX Network Reference Model (NRM)

The WiMAX network architecture, usually represented by a Network Reference Model (NRM), allows the flexibility to configure logically related functional entities between various network entities [30].

The WiMAX NRM, shown in Figure 4, has of several logical network entities: Mobile Stations (MSs), an access service network (ASN), and a connectivity service network (CSN), and their interactions through reference points R1–R8. Each MS, ASN, and CSN represents a logical grouping of functions [14]. CSN is the network entity providing IP connectivity to the WiMAX
radio equipment, CSNs are maintained by Network Service Providers (NSPs). The ASN and CSN are further broken up into reference points. The currently defined reference points are used for different control and management purposes, as well as for data bearing between the network entities [15]. The next section will further explain these entities in detail.

2.2.3.1.1 Subscriber Station (SS)/ Mobile Station (MS)

The MS is the device used by the subscriber to access the WiMAX network. This device can be used by more than one user and the same user can access the network with more than one MS [18].

2.2.3.1.2 Access Service Network (ASN)

Access Service Network (ASN) represents a boundary for functional interoperability with WiMAX clients and WiMAX connectivity services and helps to handle the layer 2 connectivity plane, [18].

2.2.3.1.2.1 Base Stations (BS)

WiMAX BS is similar to a cell phone tower and a single WiMAX BS/tower can transmit different types of network traffic like voice, video, and data signals across distances of up to 50 km from a central tower with unobstructed Line of Sight (LOS) at rates as high as 70 Mbps and it can also connect directly to the Internet [17].

2.2.3.1.2.2 ASN Gateways (ASN-GWs)

The ASN gateway acts as a layer 2 traffic aggregation point within an ASN, the ASN gateway has many other functions which will not be discussed in this study,[18].

2.2.3.1.2.3 Network Access Provider (NAP)

The NAP is the physical point used by the subscriber terminal to access the network [18].

2.2.3.1.3 Connectivity Service Network (CSN)

The CSN provides connectivity to the Internet, ASP, other public networks and corporate networks.
2.2.3.1.3.1 Authentication, Authorization and Accounting (AAA)

AAA refers to a framework based on IETF protocols, Remote Authentication Dial-in User Service (RADIUS) or Diameter for authentication, authorization, and accounting associated with the user terminal subscribed services across different access technologies [19].

2.2.3.1.3.2 Home Agent

The Mobile IP Home Agent (HA) resides in a CSN, it is responsible for maintaining MS position information and sending the packets to the network of MS [21].

2.2.3.1.3.3 Dynamic Host Configuration Protocol (DHCP)

In any network infrastructure, each times network device connects to a network infrastructure, it obtain a new local IP address through the DHCP server [20], in WiMAX network this server resides in the CSN and its function is to provide the DHCP clients with IP addresses.

2.2.3.1.3.4 Application Service Provider (ASP) Network

An application service provider (ASP) is an entity offering hosted services for subscribers. These include web hosting and data storage, backup and disaster recovering, and so on [22].

2.2.3.1.3.5 Visited NSP (V-NPS)

A visited NSP is a WiMAX service provider that a subscriber uses to access the network in a roaming scenario [18].

2.2.3.1.3.6 Home Subscriber Server (HSS)

The HSS acts the database holder of subscribers’ public and private identities, security variables, and location information [31].

2.2.3.1.3.7 Home NSP (H-NSP)

The H-NSP authenticates and authorizes subscriber sessions and is responsible of the billing and charging procedures even in a roaming scenario where the subscriber is moving through various NSPs [18].
2.2.3.1.4 Reference Points (RP)

RP is the end-point of the communication between two functional entities, it constitutes the standard interface that are used for interoperability among components of different manufacturers [18].

2.2.4 WiMAX Protocol Layers

Protocol layers are a hierarchical model of network and the divisions are referred to as layers or levels, with each layer performing a specific task, below it and performs services to the protocol layer above it as in Figure 5.

![Figure 5: WiMAX Protocol Layers](image)

These layers are actually a division of WiMAX protocol stacks which have two main layers that are MAC layer & PHY layer, MAC common part sub layer & Security sub layer is shown in Figure 5 [23].

2.2.4.1 MAC Layer

The primary task of the WiMAX MAC-layer is to provide an interface between the higher transport layers and the PHY-layer. The MAC-layer takes packets from the upper layer, these packets are called MAC service data units (SDUs) and organizes them into MAC protocol data units (PDUs) for transmission over the air [25]. These sub layers interact with each other through the service access points (SAPs) as shown in Figure 5.
2.2.4.1.1 Convergence Sub-layer (CS)

The ATM Convergence sublayer is defined for ATM services, and the packet convergence sublayer is defined for mapping packet services such as IPv4, IPv6, Ethernet, and virtual local area network (VLAN) [66]. In addition to the basic functions, the CS also performs functions such as payload header suppression and reconstruction to enhance airlink efficiency [26].

2.2.4.1.2 Common Part Sub-layer (CPS)

The CPS provides MAC functionality to support broadband applications, CPS includes many procedures of different types which includes [27] the QoS for the transmission and scheduling of data over the PHY Layer.

2.2.4.1.2.1 Security Sub-layer

Security Sub-layer is also known as Privacy Sub-layer, it provides authentication, secure key exchange, encryption and integrity control and ensures privacy services to the subscribers across the wireless network and protects from unauthorized accesses [27].

2.2.4.1.2.2 MAC Protocol Data Unit (PDU) Format

The MAC protocol data Unit (PDU) is the data unit exchange between the BS and the SS MAC layers. It consists of a fixed length generic MAC header which may be followed by a variable length payload. The flexible length of the MAC PDU enables it to deal with different types of higher traffic layers and also contain a cyclic redundancy check (CRC) [28].

2.2.4.2 Physical (PHY) Layer

The purpose of the PHY layer is to transport and receive the physical data in a wireless medium. The PHY layer frequency of operation is from 10 to 66GHz and single carrier; and it requires Line of Sight (LoS), to overcome the LoS issues, it uses the frequency spectrum between 2-11GHz [10].

WiMAX 802.16 PHY-layer considers two types of transmission techniques OFDM and OFDMA. Both of these techniques have frequency band below 11 GHz and use TDD and FDD as its duplexing technology [29].

2.2.4.2.1 Orthogonal Frequency-Division Multiplexing (OFDM)

Orthogonal frequency division multiplexing (OFDM) is a multicarrier modulation technique for high-data-rate applications, it is efficient and flexible in management of intersymbol interference (ISI) in highly dispersive channels [16]. The OFDM based communication systems transmit
multiple data symbols simultaneously using orthogonal subcarriers. The orthogonality in the OFDM ensures the separation of subcarriers at the receiver side [32], the OFDM transmitter and receiver systems are described in Figure 6 and 7 (as adapted from [32]).

![Figure 6: OFDM transmitter](image)

![Figure 7: OFDM receiver](image)

### 2.2.4.2.1.1 Frequency domain

The frequency domain representation of an OFDM symbol and its orthogonal subcarriers is shown in Figure 8 [27]. The size of the Fast Fourier Transform (FFT) point determines the number of subcarriers. There are three subcarrier types.

- **Data subcarriers**: For data transmission
- **Pilot subcarriers**: For various estimation purposes
- **Null subcarriers**: No transmission at all, for guard bands, non-active subcarriers and the DC subcarrier.
2.2.4.2.1.2 Orthogonal Frequency Division Multiple Accesses (OFDMA)

OFDMA has become the basis for several emerging broadband cellular networks, such as 802.16e (WiMAX), or 3GPP Long Term Evolution (LTE). In an OFDMA-system, many users are multiplexed in time and frequency on the basis of the underlying OFDM transmission system [33].

The frequency domain specifications are similar to OFDM PHY specifications. A two-dimensional allocation of OFDMA PHY may be visualized as a rectangle, such as the $4 \times 3$ rectangle shown in Figure 9 [27].

2.2.5 WiMAX Performance

WiMAX claims to have the high performance but due to its lack of deployment not enough off-the-field empirical data is available that would provide quantitative and qualitative attestation of the performance and operational capabilities and efficiencies of the WiMAX link [34]. The performance of WiMAX physical layer is commonly determined by Bit-Error-Rate (BER) [41]. Noise in transmission medium disturbs the information signal and causes data corruptions. In general, Signal-to-Noise ratio (SNR) is inversely proportional with BER and to improve the BER performance of a wireless communication system burst error needs to be reduced as much as possible. Forward error correction techniques use error-correcting codes such as Reed-Solomon (RS), Convolutional Code (CC), and Cyclic-Redundancy-Check (CRC) to deal with burst error and by concatenating two different codes, the total BER of WiMAX system can be improved [36].
2.2.5.1 Reed-Solomon (RS)

Reed-Solomon (RS) codes [40] are error correction codes with applications ranging from the compact disc, communications, to the exploration of the solar system. RS codes are proven to be effective in transmitting data in wireless channels [39]. Reed-Solomon error correction works by first constructing a polynomial from the data symbols to be transmitted, and then sending an oversampled version of the polynomial instead of the original symbols themselves [76].

2.2.5.2 Bit Error Rate (BER)

At high vehicular speeds, the WiMAX in mobile communication rapidly changes in surrounding environments and cause severe fading at the receiver and that causes a drastic fall in throughput unless proactive measures are taken into account, throughput becomes insufficient to support many applications, particularly those with multimedia contents. Bit Error Rate (BER) estimation is an integral part of any proactive measure and recent studies suggest that Nakagami-m model performs better for modelling channel fading in wireless communications at high vehicular speeds [41].

2.2.5.3 Cyclic-Redundancy-Check (CRC)

The CRC is an error detection method used as a part of transmission techniques for error detection of the payload when the error detection [42]. The receiver verifies that the packet was received with errors using the CRC code that is attached to each and every information block, then the receiver sends either a positive acknowledgement (ACK) or a negative acknowledgement (NACK) concerning the reception of the data block [43].

2.2.5.4 Signal to noise ratio (SNR)

SNR is defined as the ratio of the desired signal power to noise power. SNR indicates the reliability of link between the transmitter and receiver and some part of spectrum is more affected by interference than the other parts of spectrum [44].

2.2.6 Quality of Service in WiMAX Networks

The term Quality of service refers to the probability of the telecommunication network meeting a given traffic contract and here it is the probability of a packet successfully passing between two points in the network. The primary goal of a good QoS is to provide priority including better throughput, controlled jitter and latency and improved loss characteristics. [35]

2.2.6.1 Delay

Delay or latency is the time taken by the packets to reach from source to destination. In end to end delay the packet losses some energy as well in the form of noise which is also taken into
consideration. End to end delay could be measured as the difference of Packet arrival and packet start time [37]. Equation 1 shows the calculation of average end to end delay which is the measure of elapsed time taken during modulation of the signal and the time taken by the packets to reach from source to destination.

\[ \text{Delay} = \sum (\text{Packet Arrival}_i - \text{Packet Start}_i) \]  

\hfill (1)

2.2.6.2 Throughput (TP)

Throughput is measure of number of packets successfully delivered in a network. It is measured in terms of packets/second. The value of throughput should be high or else it affects every service class defined in WiMAX [35]. Equation 2 shows how to calculate throughput.

\[ TP = \frac{\sum \text{Packets Delivered}}{\sum (\text{Packets Arrival}_i - \text{Packet Start time}_i)} \]  

\hfill (2)

2.2.6.3 Packet Loss or Corruption Rate

Packet loss denotes the number of data packets that are lost in transmission from the sender to the receiver and is measured as the percentage of packets that are lost as a proportion of the total volume of packets [32]. Equation 3 shows the simple equation to calculate packet loss.

\[ \text{Packet Loss} = \frac{\sum \text{Lost Packet Size}_i}{\sum \text{Packet Size}_i} \times 100 \]  

\hfill (3)

2.2.6.4 Jitter

Jitter or delay variation is the variation in the delay introduced by the components along the communication path. It is the variation in the time between packets arriving. Measuring jitter is important element that determines the performance and the QoS the network offers [38]. Equation 4 shows how to calculate jitter or delay.

\[ \text{Average Jitter} = \frac{\sum |(\text{Packet Arrival}_{i+1} - \text{Packet Start}_{i+1}) - (\text{Packet Arrival}_i - \text{Packet Start}_i)|}{n-1} \]  

\hfill (4)
2.3 Markov Chains

2.3.1 History of Markov Chains

Markov chains were introduced by Andrei Andreevich Markov (1856–1922) and were named in his honour [45]. Markov chains first appear in his work in a paper of 1906 [46] and is a mathematical model of a random phenomenon evolving with time in a way that the past affects the future only through the present. The “time” can be discrete (i.e. the integers), continuous (i.e. the real numbers), or, more generally, a totally ordered set [49]. There are three important elements in Markov chains [67]:

- Probability transition matrix $P$.
- Transition diagram.
- Steady-state vector $\pi$

Discrete-state models have proved to be a valuable tool in the analysis of these computer systems and communication networks. To model these systems one needs to describe the behaviour of the system by the set of different states the system may occupy, and identifying the transition relation among the various states of the system [50].

When the probability distributions is geometric, the stochastic process can be modelled as a Discrete Time Markov Chain (DTMC) and if the probability distributions is exponential it is modelled as Continuous Time Markov Chain (CTMC) (further details in section 2.3.5 and section 2.3.6 respectively). A Markov Decision Process (MDP) admits a number of discrete probability distributions enabled in a state which are chosen non-deterministically by the environment [51].

2.3.2 Markov Model

The term “Markov model” [52], refers to mathematical models in which the future state of a system depends only on its current state. The “Markovian property” is the memoryless characteristic which implies that all transitions from one state to another occur at constant rates, in reliability analysis the transitions usually consist of failures and repairs [52]

2.3.3 Markov Property

In this process, a set is defined as $\{X(t), t = 0,1,2,...\}$ whose number of elements is finite and denoted with real positive numbers. $X(t) = i$ represents that the process is in state $i$ at an instant of time $t$ and take discrete values. It is said that there is a fixed probability $p_{ij}$ that chain goes from a
state \( i \) to a state \( j \) in the next time \( t \). This process is represented in Equation 5 and it is known as the Markov property [67].

\[
P \left[ X(t_{k+1}) = x_{k+1} | X(t_k = x_k, \ldots, X(t_1) = x_1 \right] = P \left[ X(t_{k+1}) = x_{k+1} | X(t_k) = x_k \right]
\]

Where,

\( X(t_k) \), Current sample.

\( X(t_{k+1}) \), Future sample.

\( X(t_1), \ldots, X(t_{k-1}) \), Past samples.

\( x_k \), State of the sample in the moment \( k \).

Figure 10 shows the system that can change from one condition to another or system that continues in the same state [67].

![Figure 10: Markov Property](image)

### 2.3.4 Generative Channel Models

The generative channel model is a Markov chain with a finite or infinite number of states with defined transition probabilities and with this we can produce Markov state sequences, and in this way simulate the transitions in real channel behaviour from one state to other states [58]. In a generative approach, a model for some particular task is learned from the joint probability \( P(x, y) \) of inputs \( x \) and output labels \( y \). Predictions are made by using Bayes Rule to calculate \( p(x|y) \) and then choosing the most likely \( y \). This approach allows generating the observable data and permits the exploration of the generation process of these data [57].
There are some widely used Markov models which represent channels with memory, including the Gilbert model [48], the Gilbert-Elliott [55] model and Fritchman's [56] partitioned finite-state model.

2.3.4.1 Gilbert's Model

Gilbert introduced his model in [48] to model the errors that occur on telephone networks that transmit digital data [49]. The state transition diagram of Gilbert's model is shown in Figure 11.

![Figure 11: Gilbert's Channel Model](image)

Gilbert initiated the modelling of channels with memory [53] when he proposed the use of finite-state Markov models in 1960. He proposed a two-state Markov chain with a good state and a bad state [54]. The purpose of Gilbert's scheme, in using the two state Markov chain, is to make the probability of generating errors variable. This makes the Markov chain a more flexible way of modelling the real channel [58]. The Gilbert model does suffer from some limitations with respect to its ability to successfully model some real channels in practice. The limitations arise from its renewal nature and from the assumption of the run lengths of G and B [59].

2.3.4.2 The Gilbert-Elliott model

The estimation of quality for real-time services over telecommunication networks requires realistic models for impairments and failures during transmission [60]. Since the Gilbert model has limitations, Elliott [55] suggested a modification to Gilbert's model to assume an error probability in the good state G of the Gilbert model, this model is called the Gilbert-Elliott model [58]. Due to the underlying Markov nature of the state process, the channel has memory that depends on the transition probabilities $P$ and $p$ between states [61], as shown in Figure 12.
Elliott introduced another parameter $k$, the probability of correct reception of a bit when the channel is in the good state. Hence, the error sequence is generated not only by one binary symmetric channel but by two binary symmetric channels. This makes the generation of the error sequence more flexible, and the model now becomes a non-renewal Markov chain [58].

2.3.4.2.1 Simple Partitioned Channel Model

The simple partitioned channel model is a finite state Markov chain with $N$ states, partitioned into $N-1$ error-free states and a single error state, with transitions between states as shown in Figure 13 [54].
2.3.5 Continuous Time Markov Chains (CTMC)

Continuous Time Markov chains (CTMCs) [50] are used for the performance analysis of computer systems and it typically require the generation and storage of the underlying state space of the CTMC, and the processing of the state space for the numerical solution. CTMC models for even trivial real-life systems are usually huge, and both the amount of required memory and the time to compute the solution pose a major difficulty [50].

2.3.6 Discrete Time Markov Chains (DTMC)

A discrete-time Markov chain is a stochastic process which is the simplest generalization of a sequence of independent random variables [68], it is a process whose state space is a finite or countable set, and whose (time) index set is \( T = \{0, 1, 2, \ldots\} \). Discrete time Markov chains are widely used to model systems with uncertain information [70]. A DTMC model consists of a list of the possible states of the system along with the possible transition paths among these states [72]. This model is the most widely used stochastic process for analysing the reliability, maintainability and safety of real-world applications [71].

2.3.7 Elements of Markov chains

A Markov system (or Markov process or Markov chain) is a system that can be in one of several (numbered) states, and can pass from one state to another each time step according to fixed probabilities[74]. There are three important elements in Markov chains [67]:

- Probability transition matrix \( P \).
- Transition diagram.
- Steady-state vector \( \pi \)

2.3.7.1 Probability transition matrix

A transition matrix has several features [75]:

- It is square and all possible states must be used both as rows and as columns.
- All entries are between 0 and 1, inclusive.
- The sum of the entries in any row must be 1.

The switch between states is established in the probability transition matrix \( P \). Each element represents the probability that switches or remains in the state. These switches are called transitions. \( P \) is a square matrix whose order is the same to the number of states [67]. Equation 6 shows the structure of a probability transition matrix.
Each element of $\mathbf{P}$ must satisfy the next condition.

$$p_{ij} \geq 0, \quad i,j = 0,1,2,3\ldots$$

Where,

$p_{ij}$, Probability that current sample is in state $i$ given the immediate predecessor past time was in state $j$.

$i$, Transition state $i$.

$j$, Transition state $j$.

### 2.3.7.2 Transition diagram

The graphical representation of a Markov chain is a transition diagram, this is equivalent to its transition matrix [73]. The transition diagram is also used to describe the model definitions [67]. Figure 14, Figure 15 and Figure 16 show probability transition diagram for the 2-state, 3-state and 4-state Markov chain respectively.

![Transition Diagram](image)
2.3.7.3 Steady-state vector

Another important element in Markov chains is the steady-state vector $\pi$, which represents the total appearing percentage of a state in a Markov chain. This vector can be computed by raising $P$ to a large power [67], this is shown in Equation 8.

$$P^n \rightarrow 1\pi \quad (8)$$

Where,

$P$, Probability transition matrix.

$\pi$, Steady state probability vector

$1$, Column vector of ones: $1^T = (1,1,...)$
One property of the $\pi$ vector is that the sum of its elements must be equal to one \([67]\), as it is shown in Equation 9.

$$\sum_{1}^{\pi_i} = 1$$  

(9)

Where, $\pi_i$ is a steady state probability for state i.

### 2.4 Simulation

Simulation is a key networking tool in research due to its essential advantages over testing with real hardware \([77]\). Issues related to budget and cost of time are some of advantages that make simulation attractive for research, while flexibility, scalability and virtually no cost expansion are also valuable advantages over other testing methods \([78]\). Network simulators offer a high degree of flexibility to test new protocols, technologies, conceptual models and topologies \([79]\). The main motivation behind network simulation is to achieved high level of reliability and minimise maintenance costs as much as possible when developing a new technology \([87]\). The proposed simulation software for this research is NS-3 and OPNET, in this section the research is about the in-depth study these two simulators.

#### 2.4.1 Network Simulator-3 (NS-3)

NS-3\([86]\) is a free, open source software project building and maintaining a discrete-event network simulator \([80]\). Events are actions such as sending a packet, and running the scheduled event until completed without advancing the simulation time, and then the simulation time is increased to the start time for the next scheduled event \([81]\). Before NS-3 was developed, the NS-2 simulator \([83]\) was a widely used simulator for research and education on Internet and other network systems. Borrowing concepts and implementations from several open source simulators including NS-2, yans \([84]\), and GTNetS \([85]\). NS-3 is written in C++ with a highly flexible architecture. It allows third party contributors to design new models and incorporate them into the main NS-3 code, resulting in a continually increasing scope. Components such as WiFi and WiMAX standard implementations, mobility models and routing protocols, make NS-3 well suited for wireless network simulations \([86]\).
2.4.1 NS-3 Distribution

NS-3 is an all-in-one installation package that consist of NS-3 source code that is ready to compile and the NS-3 compilation scripts inform about the missing linux based software dependencies that can be downloaded from linux repository [80].

2.4.1.2 Integration with Python

NS-3 supports Python [115,116] during scripting and when visualization tasks are initiated, also a set of bindings is generated to allow interaction of a Python script with the NS-3 API which is accessible from a C++ script [81]. Python is a high level interpreted and interactive programming language. The programming language was designed by Guido van Rossum in 1990 [115]. Python also provides high-level data structures [116] and it can be said that it is an object-oriented programming language.

2.4.1.3 Realism

NS-3 uses nodes to emulate real computers [81] and by that it easily creates realistic network topologies, this includes support for interfaces such as sockets and network devices, multiple interfaces per nodes, use of IP addresses, and other similarities [83].

2.4.1.4 Performance

NS-3 offers superior performance in computation and memory footprint compared to NS-2 [82]. The source of the memory footprint gains are straightforward, and the aggregation system prevents unneeded and sometimes very large parameters from being stored when they are unnecessary, and packets do not contain large amounts of meta-data and unused, reserved header space [81].

2.4.1.5 Simulation Output

Animation / simulation tool shows the network topology and the data flow through that topology. This is displayed as the simulator is running or after the fact from a trace file generated by the simulator [81]. NS-3 initialises PyViz, which is a python based real-time visualization package and it takes input directly from trace hooks in the simulation as it runs [81].

2.4.2 OPNET

OPNET [89] is a proprietary network simulator [87] and a software package that supports general network modelling, it is mainly used for particular types of network simulation projects [89]. OPNET simulations are based on discrete events like NS-3 and one disadvantage of OPNET is the fact that it is proprietary software and therefore limited in terms of customizability [87]. A large range of communication systems from a single LAN to global inter-networks is supported [90], also OPNET includes a vast library of communication devices, media, and protocols [91].
2.4.2.1 OPNET Packages

OPNET software products include OPNET IT Guru and OPNET Modeler and these are the most popular for educational use.

2.4.2.1.1 OPNET IT Guru

OPNET IT Guru is an easy to use and easy to learn network simulation package which allows a user to use point-and-click to create and configure simple or sophisticated network systems, and can conduct simulations to study and analyze the performance of the system [91].

2.4.2.1.2 The OPNET Modeler

The OPNET Modeler includes model design, simulation, data collection, and data analysis also provides a comprehensive development environment supporting the modelling of communication networks and distributed systems. Both behaviour and performance of a model are analyzed by performing discrete event [92].

The project editor helps in graphical representations of communication network topology and the node editor enables the editing of network device models. The standard model contains visualization of communication devices with all protocol layers according to the ISO/OSI communication model [94].

The process editor is intended for process modelling, every state in the process model includes C/C++ code, supported by an expanded functions library, intended for programming communication protocols [96].

2.4.2.1.3 OPNET Graphical User Interface

A Graphical User Interface (GUI) supports the configuration of the scenarios and the development of network models. Three hierarchical levels for configuration are differentiated: i) the network level creating the topology of the network under investigation, ii) the node level defining the behaviour of the node and controlling the flow of data between different functional elements inside the node, and iii) the process level, describing the underlying protocols, represented by finite state machines (FSMs) and are created with states and transitions between states [93].

2.4.2.1.4 OPNET Modeler Wireless package

Wireless links are simulated by using an open concept – ‘transceiver pipeline’ that enables delay computations during the spread of radio waves, closes radio links, considerate of an aerial’s emissive diagram, modulation effects, bit-error rate, interference, forward error corrections,
background noise, etc. This concept considers effects caused by field influences over the modelled terrain, such as fading, reflections, diffraction, atmospheric absorption etc. [94].

2.4.2.2 Simulation Time

Time to run the simulation, generate reports and statistics in OPNET varies as the numbers of sensor nodes increase or decrease. It is important to understand that increasing the number of nodes will also increase the running time. This increase in simulation time starts slowly when the number of nodes is low then takes an exponential manner as the number of nodes increase [92].

2.5 Similar Work

WiMAX has been around for some decades but it is not yet fully implemented and still many people do not know about it except those engaged in its study or research. There is much information related to WiMAX and studies related to its performance simulation using different simulation programs such as OPNET, NS-2, NS-3, MATLAB [114] and others. The modelling of WiMAX using Markov Model has also been researched and MATLAB, has been a simulator of choice in this regard. In this section we will look at the similar studies related WiMAX performance simulations.

2.5.1 Simulation of WiMAX with NS-3

In [81] it is stated that in the NS-3 simulator there is a core consisting of a scheduler and several useful classes defining nodes on a network, packets, and other similarly near universal concepts. In [81] authors also state that NS 3 is much more useful when trying to analyse network performance, as the vast array of tools available to analyse real networks are generally available. In [97] the authors elaborate WiMAX Module for the NS-3 and presents a detailed design and implementation of an IEEE 802.16 WiMAX module for the NS-3 and have listed the key parameters of OFDM PHY. In [98], the authors present detailed design and implementation of features that enhance the WiMAX module of the NS-3 simulator.

2.5.2 Simulation of WiMAX with OPNET

In [99] the authors introduced a new distributed model that improve the performance of QoS over fixed WiMAX with respect to parameters that include throughput, delay and application response time, to enhance the services that are provided to the end users, they have minimized delay, application response time and increased throughput. They developed a model based on their proposed new distributed model (Master-Slave), the first scenario comprised of 3 Base Stations (BS) and thirty Subscriber Station (SS) with one Master BS which is selected by the designed algorithm (Nearest Neighbourhood Algorithms). The design has been evaluated using the simulation tool OPNET Modeler 16.0 and the results shows significant increase in the network
throughput and drastic decrease in network delay and application response time. In [100] authors do performance analysis of physical layer of WiMAX system using OPNET modeller.

2.5.3 Modelling WiMAX PHY Layer with Markov Chain

In [100], the author analyses the WiMAX number of polls based on the Markov chain analysis to minimize average polling delay while increasing network throughput. The author used the continuous-time Markov model to evaluate the delay and throughput of contention-free unicast polling. In [101] the conventional Markov model was implemented to analyse the traffic class queue model. The Markov model was constructed for the packet queue. The author also specified parameters that will be used when simulating with OPNET. In [102] the author proposed a two-dimensional Markov chain model to model the saturated IEEE 802.16 network.
3 RESEARCH METHODOLOGY

3.1 Introduction

We use simulation, and recent technological advances have enabled discrete event simulation to be utilized, simulation is now being chosen for execution on distributed networks, and multiprocessors, as well as sequential architectures. Simulation is no longer simply a tool for analysis but it is expected to provide support for a wide variety of purposes including, training, interaction, visualization, hardware testing, and decision support in real-time, etc [106].

In [107], the author mentioned eight major phases for the proper application of simulation methodology, namely:

Phase 1. Define-the problem
Phase 2. Design the study
Phase 3. Design the conceptual model
Phase 4. Formulate inputs, assumptions, and process definition
Phase 5. Build, verify, and validate the simulation model
Phase 6. Experiment with the model and look for opportunities for design of experiments
Phase 7. Document and present the results
Phase 8. Define the model life cycle

This chapter presents the methods and procedures that will be used in the research, which is the modelling of a WiMAX using Markov Model, the simulation of a WiMAX System using the NS-3 simulator, the simulation of a WiMAX System using the OPNET simulator and the analysis of simulation results.

3.2 Approach and Design

Simulation is the best bet considering the fact that WiMAX is not cheap. Smit in [104] states that most networks may consist of a number of elements and to perform measurements on such a large network at every node and on every stream is practically impossible. For this reason, many researchers only focus on end-to-end measurements and that is a call for using simulation. Simulation at packet level consists of data pertaining to a particular simulation scenario, the data contains information about generated, lost, dropped and queued packets. These all also contain timestamps, which enables one to make accurate end-to-end delay and other types delay measurements. Using simulation, the detailed statistics about the processes that occur in the network can be easily measured. The advantage of simulation is that the results can be produced repeatedly, and that enhances the analysis and comparison of different results.

In this research a performance will obtained by increasing a number of Subscriber Stations and simulate as we increase the SS number. The simulation will first be done with Twenty (20) SS's then these SS's will be incremented to 50 then 75 and lastly 100 SS's.
3.2.1 Topology
The Network Topology that will be used is shown in Figure 144, where the network consist of one WiMAX Base Stations (BS) which is connected to the Internet gateway. The BS will consist of the Subscriber Stations (SS's) of up to 100, at first, 20 SS's will be connected then incremented to 50 and then increment to 75 and lastly to 100.

3.2.2 Protocols and Applications
The type of protocols to be used to be used is the UDP and the TCP with the intention of analysing the performance of WiMAX as the SS's are incrementated. The simulation will monitor the download of text, audio and video data over WiMAX network.

3.2.3 Expected Simulation Outcome
The expected simulation output to analyse is the following:
- Delay
- Throughput

3.2.4 Simulation Parameters
For traffic configuration, both SS nodes had an uplink application load of 20 Kbps for a total of 0.4 Mbps. SS nodes will be configured to use QPSK, 16 QAM and 64 QAM for both uplink and downlink applications to measure the characteristics. The BS will use an SOFDMA frame with 512 subcarriers and frame duration of 5 milliseconds. Simulation time is taken as 360 seconds. The parameters that will be used for both NS-3 and OPNET modeller are shown in TABLE 3 for Subscriber Stations parameters and in TABLE 2 for Base Station parameters.
### Table 2: Parameters for the Subscriber Stations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antenna Gain (dBi)</td>
<td>-1 dBi</td>
</tr>
<tr>
<td>Max Power Transfer (W)</td>
<td>0.5</td>
</tr>
<tr>
<td>PHY Profile</td>
<td>Wireless OFDMA 20 MHz</td>
</tr>
<tr>
<td>PHY Profile Type</td>
<td>OFDM</td>
</tr>
<tr>
<td>Modulation and coding</td>
<td>QPSK, QAM 16, QAM 64</td>
</tr>
<tr>
<td>Pathloss Model</td>
<td>Free Space</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>64 KB</td>
</tr>
<tr>
<td>Terrain Type</td>
<td>Terrain Type A</td>
</tr>
</tbody>
</table>

### Table 3: Parameters for the Base Stations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antenna Gain (dBi)</td>
<td>15 dBi</td>
</tr>
<tr>
<td>Maximum Number of SS nodes</td>
<td>100</td>
</tr>
<tr>
<td>Maximum Power density (dBm/subchannel)</td>
<td>-110</td>
</tr>
<tr>
<td>Ranging Backoff Start</td>
<td>2</td>
</tr>
<tr>
<td>Ranging Backoff End</td>
<td>4</td>
</tr>
<tr>
<td>PHY Profile</td>
<td>Wireless OFDM 20 MHz</td>
</tr>
<tr>
<td>PHY Profile Type</td>
<td>OFDM</td>
</tr>
</tbody>
</table>

### 3.3 Simulation with NS-3

WiMAX Performance Simulation using NS-3 will be done in Ubuntu Linux, employing the design explained in section 3.2, the choice of this operating system for NS-3 is because the simulator works best in Linux and also information for both NS-3 and Linux is freely available with a huge online community that assists in installation and basic use of the simulator. The Ubuntu Linux flavour is chosen because of wide support base and ease of NS-3 installation.

A similar work has previously been done by the authors in [11], where they had a WiMAX network scenario of 20 number nodes and sub nodes with two base stations, base station 1 and base station 2. The across Base station 2 (BS2) there are total 10 number of nodes out of that 6 are Subscriber Nodes (SN) and another 4 are subscriber sub nodes (SSN). All are fixed nodes. Across BS1 there are total 10 numbers of nodes out of that 6 are SN and another 4 are SSN and all nodes are mobile nodes. Here the author simulates the above mentioned network scenario for analysis of the WiMAX performance. In this research the WiMAX network is also simulated and analysed, but here we the number nodes is increased by the number stated in section 3.2 and then analysis will be done by comparing the performance as we increase the nodes.
To simulate using NS-3, the following steps will be followed:

- Converting the nodes of a network model to a C++ program.
- Running the program in Command Line Interface (CLI) using waf command of NS-3
- Run Simulation
- Plotting Graph of the simulation output in CLI Gnuplot [115]
- Analyse the output graph

A division of the simulation task for any evaluation of performance into many steps is helpful, researches had divided these simulation steps into 8 and some to 12 steps [47]. For example in [47], the researcher have divided the simulation into 8 steps as shown in Figure 17 below.

![Figure 17: 8 Steps of Simulation](image)

3.3.1 NS-3 Simulation Reliability

The authors in [62] has worked with the NS–3 team to help them appropriate default parameters, they state that adaption of the parameters can help to gain more realistic behaviour. They also noted that there are models accounting for the effects in different scenarios, for example the urban and free open space, these feature more realistic behaviour than abstract models. In the NS-3 model library models are defined as abstract representations of real world objects, protocols, devices, etc. [87]. The authors of [63] are comparing the performance and behaviour of the IEEE 802.11 MAC model of NS-3 to a test bed providing the same functionality. They use different scenarios with different focuses and objectives and then compare their measurements. They concluded that the IEEE 802.11 MAC model is a good representation of reality [87] by using NS-3.
The NS-3 802.11 Implementation in [69], the author states that NS-3 Packet objects contain a buffer of bytes in which protocol headers and trailers are serialized. The content of this buffer has to be the same as the content of a real packet on a real network in the desired scenario. The design of this object has been done in order to:

- Avoid changing the core, with different kind of packets.
- Maximize the integration with real-world code and systems.
- Make it easy for the fragmentation, defragmentation and concatenation processes.
- Make efficient use of memory.

Another author in [95] has presented a paper about validation study of the IEEE 802.11 MAC model in the NS-3 simulator using the EXTREME testbed, their study showed that there is a good qualitative agreement between a simulation and a testbed and have also a case where there are noticeable quantitative differences. These differences show that they are not always caused by the simulator but also the particular choice and configuration of the devices in the testbed can play an important role.

3.4 Simulation with OPNET

OPNET modeller is used to simulate the WiMAX performance for the explained topology in section 3.2 and it runs on Windows 7. In [24] the authors have done a similar study of Simulation and Performance Evaluation of WiFi and WiMAX using OPNET, they use OPNET Modeler to simulate and compare WiFi and WiMAX in a small area network and compare their performance in terms of mobility. In this research we simulate WiMAX performance by increasing the number of SS's as explained in section 3.2, here we compare the performance as we increase the number of SS's in OPNET.

The following are the necessary steps for the WiMAX network to simulate over the OPNET modeller [105].

- Create the initial technology
- Create the wireless development scenario
- Add traffic to wireless network mode
- Configure BS, SS and WiMAX parameters.
- Running and analysing simulation results

The steps above can be further described as below.

→ **Create Network Model:** Here you create a new project, topology, server and network devices and traffic. You also configure network size and links between hosts and network devices.

→ **Choose Statistics:** Here you create traffic load for hosts in bps, and network delay in seconds.

→ **Run Simulation:** Here you run the created project.

→ **View and Analyse Results:** Here you can view the plotted results for analysis.
3.4.1 Accuracy of OPNET Modeler

OPNET contains a vast amount of models of commercially available network elements, and has various real-life network configuration capabilities. This makes the simulation of real-life network environment using OPNET Modeler close to reality [88]. The authors in [95] compared the accuracy of network simulators for packet-level analysis using a network testbed, the comparison was between the NS-2 and OPNET. They noticed that the complete set of OPNET Modeler modules provides more features than Ns-2, and it therefore will be more attractive to network operators.

3.5 Markov Modelling of WiMAX

We also model the Topology with Markov Chains. The number of served SSs is an important parameter that significantly affects the network performance. The method used for Markov Modelling is similar to that used in [105], where the author stated that to model the arrival process and the traffic source we use the Batch Markov Arrival Process (BMAP), which enables more realistic and more accurate traffic modelling. They determined analytically different performance parameters, such as average queue length, packet dropping probability, queue throughput and average packet delay.

The Network Topology that will be used is the one shown in Figure 144 and also the parameters will be the same as those specified above in Table 2 and Table 3.
4 SIMULATION USING OPNET MODELER 14.5

4.1 Introduction

We do performance simulations of WiMAX using Markov model and simulators. We have decided to start with simulation using simulator, and that is OPNET Modeler 14.5 and Network Simulator 3 (NS-3). In this chapter we will deal with the simulation using OPNET Modeler 14.5. The installation of the OPNET Modeler 14.5 was done on the following platform.

Manufacturer : Hewlett-Packard
Model : HP EliteBook 2570p
Processor : Intel Core i5, 2.50GHz
Installed Memory : 4.00 GB
System Type : 64bit Windows 7 Professional

For each scenario a new project will be created, for this research we will have four (4) project as follows:

- Simulation of twenty (20) Subscriber Stations
- Simulation of fifty (50) Subscriber Stations
- Simulation of seventy five (75) Subscriber Stations
- Simulation of hundred (100) Subscriber Stations

The simulation steps that we will follow are defined in the previous chapter which is Methodology and outlined the following steps:

- Create Network Model
- Choose Statistics
- Run Simulation
- View and Analyse Results

4.2 Starting the OPNET Modeler 14.5 Simulation Program

Before we can start simulation, the OPNET program needs to be installed and simulation environment variables need to be thoroughly set up, these settings can easily be found on internet with ease and we will not discuss them on this research. This section will show in details, steps to start OPNET Modeler 14.5 on Windows 7.

1. After powering up a Windows 7 computer and logging in, click on the Windows Start button (On the bottom left corner of the computer screen). Then select All Programs as seen on Figure 18 below.
2. **Figure 18: Windows Start Button and All Programs**

After selecting and clicking on All Programs, a list of installed programs will appear, scroll down and click on OPNET Modeler 14.5 as seen on Figure 19.

![List of All Programs installed in the Computer, including OPNET Modeler](image)

3. After selecting and clicking on OPNET Modeler 14.5, a list of related OPNET Modeler 14.5 programs will be displayed, see Figure 20.
4. Select the program OPNET Modeler 14.5, the last one on the sub programs. This will bring up two windows as seen in Figure 21, then click on “I ACCEPT” button. **NOTE:** Do not select **OPNET Modeler 14.5 (64-bit).**
5. After clicking the “I ACCEPT” button, one of the windows (License Agreement window) will disappear and leave one window (main OPNET Modeler 14.5 Window) as shown in Figure 22.

![OPNET Modeler 14.5 Main Window](image)

Figure 22: OPNET Modeler 14.5 Main Window

4.3 Simulation for Twenty (20) Subscriber Stations (SS)

In this section we will create a new project for WiMAX Simulation with 20 SS, the parameters will be same as those described in Methodology Chapter.

4.3.1 Creating Network Model

In this step we create a new project which is named WiMAX20SS. This stage describes the number of Subscriber Stations in our WiMAX network. The Wireless topology is chosen as this contains the WiMAX Topology and create the environment to simulate WiMAX Infrastructure which contains 20 SS, 1 BS, the remote applications server (the applications are HTTP server, video conference server and a voice server) and the backbone which represents the device that routes to internet (e.g. a router). The radius of the network, the link from the Base Station to the backbone and a link from the backbone to the server are also configured.

The steps for creating a Network Model are as follows:

1. After starting the OPNET Modeler 14.5 (as detailed in section 4.2), Click on *File* and select *New* then click OK, this takes you to the screen shown in Figure 23.
2. When on window shown in Figure 23, change the Project Name. The project name is WiMAX20SS, this name is chosen according to the description in the Methodology that we will first simulate using twenty (20) Subscriber Stations then fifty, etc., as described in the Methodology Chapter. After changing the name we click OK button, this will takes us to another window as shown in Figure 24.

3. By clicking “Quit” button (shown in Figure 24), the program quits the Wizard for Initial Topology, this allows us to create our own custom topology.
4. To create our own Network Model, we click on Topology then choose and click Deploy Wireless Network, as shown in Figure 26.

5. Selection from above in Figure 26 will take us to the Wireless Network Deployment Window shown in Figure 27, this where we start to deploy the wireless network.
From the previous window, Figure 27, we select continue and this will take us to the Wireless Deployment Wizard Window, shown in Figure 28.

6. On the Wizard in Figure 28, we select the radio button to use wizard to provide network specification. This takes us to a window that configures the Location, shown in Figure 29.
7. In Figure 29 we select “create network to select subnet” and also select the location of the WiMAX network by changing the Coordinate of the Latitude and Longitude. Here we leave default coordinates as they are. When clicking Next button, this takes us to the Technology specification windows.

8. In Figure 30 we choose desired Wireless Technology by selecting in a drop down menu of Choose Technology and clicking on WIMAX, we also change the Pathloss and Multipath Model from Vehicular to Pedestrian as this study focuses on fixed SS as seen in Figure 31.
9. After clicking the next button it takes us to the Topology window (in Figure 32). Here we choose the number of cells, the cell radius and Place nodes. We have changed Number of Cells to 1 and radius to 10 km, also the Geographical Overlay was changed to Cell (Hexagon) and Place nodes in Circular. Then click next.
10. The next window takes us to the configuration of cells, shown in Figure 33, here we specify the number of SS in a cells (For this part 20 SS's were selected) and Node Name Prefix. In this study, the prefix for Base Station is BS and SS for Subscriber Station. The tick box at the bottom of this windows must be ticked to make sure the BS is connected to the backbone, then Click the Next Button. Note that a subscriber station for this study is a workstation (Laptop or Desktop with WiMAX adaptor, WiMAX adaptors can be purchased from computer shops).

![Figure 33: Topology Configuration](image)

11. On this windows (Figure 34), the Node Mobility window, the speed is by default set to 5m/s and we have to ensure that the number of nodes is 20.

![Figure 34: Node Mobility Window](image)

12. The next window is the Summary window which displays the summary of the Network Model and the configurations (see Figure 35), when we click the finish button we get WiMAX Infrastructure with a Base Station (BS) and twenty (20) Subscriber Stations which are within a 10 km radius of a BS as we have configured (see Figure 36).
13. Note that in Figure 37 the server is not in place, we therefore will add a server that will host applications for simulation which are Text, Voice and Video. The text application will be represented by web application, where a user connects to internet for web surfing, accessing websites that will contain text and images. The websites download remote site contents onto the surfers' computers therefore the user will read the text information on the local computer but when they click another link they initiate another connection for downloading the information to their local computers again.

14. To add a server we click on the Object Pallete Button on the main menu and select the Ethernet Server (Figure 38), and in Figure 39 a link of 10Gb is selected and added between a backbone and server.
Figure 37: Selecting Application Server

Figure 38: Selecting a link for Application Server to the Backbone
15. After adding a link, we choose an option to automatically add link and ports, by choosing this option, the default backbone device is replaced by a router which in practical is used as a device that routes internet traffic (see Figure 40).

In Figure 40 we see a complete WiMAX Network Model, this will be used to simulate the performance of the WiMAX for 20 Subscriber nodes.

16. The OPNET Modeler requires that we insert configuration nodes, for this study we will insert Application Configuration, Profile Configuration and the WiMAX Configuration. The presence of these nodes is important to the OPNET Simulation. To add these nodes we again open the Object
Pallete (circled in red in Figure 41) and under the “search by name box” we write WiMAX then we click on the button next to (left hand side of) “search by name” text, ten we drag and drop these three configuration icons onto our simulation space as seen in Figure 41.

![Figure 41: WiMAX Network Model for 20 SS with Configuration nodes](image)

Figure 42 is the final WiMAX Network Model for the simulation of the performance of the twenty (20) Subscriber Stations (SS). At this stage we are ready to configure all nodes using the parameters specified in the Methodology Section. You can also re-arrange the nodes according to your liking.

![Figure 42: Final WiMAX Network Model for 20 SS](image)

### 4.3.2 Choosing Statistics

In this step we will configure the Subscriber Stations, the Base Station, the Application Server and also configure applications that were discussed in the previous chapters; that is Text, Audio and Video. The configuration of text will be based on the HTTP traffic, which is browsing of internet as this downloads text and pictures from the internet. The Audio will be represented by voice and the Video will be Video Conference.

Below are the steps we have taken to configure the nodes, there is no specific order of configuration in this step, you can start with any node, but is advisable to start with the Configuration nodes as these variables will be required by some of the nodes:
To configure the device, you need to right click on the device you want to configure and choose Edit Attributes. We will start with the Application Definition node. Here we changed the name to Application Definition then expand the Application Definitions Attribute and changed the number of rows to three (3), this 3 is for the applications we will use in this study. For each row we changed the name and the description.

**Row0**: Name → Web, Description is HTTP → image browsing

**Row1**: Name → Voice, Description is Voice → IP Telephony

**Row2**: Name → Video, Description is Video Conferencing → High Resolution Video

We have chosen Image browsing for Web, this type of browsing is heavier than normal text, but because some documents downloads include images. Therefore the image browsing is appropriate. The summary of this configuration is seen of Figure 43 and the detailed configuration for each Application is seen of Figure 45.

The next to configure is the Profile Definition, here we configure the applications profiles. The profiles are linked to the applications defined in the Application Definition. The procedure to configure these profiles is similar to that of the previous step, which is Step 1 above. The summary and the detailed configuration is seen on Figure 46.
Next we configure the WiMAX Configuration node, the summary and the detailed configuration is seen on Figure 46, Figure 47 respectively.

The figure below (Figure 48) is the detailed configuration of WiMAX Config Node. Note the rows of MAC Service Class Definitions, these are preconfigured to accommodate different types of traffic. OFDM PHY rows all have similar configurations.
Configuring SS's it is important that all are selected and right click on one to edit attributes, when saving the configuration click on check boxes “Apply to selected objects” and “Exact Match”. This helps to ensure that all similar nodes are changed to have similar attributes or configurations as the selected edited node (see Figure 48). In Figure 48 we have edited the parameters to be according to those specified in Methodology, note that WiMAX parameter for SS’s for all rows in the service class are Gold, the Gold service class is the one we configured in the WiMAX. Gold is for real time and the real time configuration provides better quality for most traffic. We there choose Gold for all traffic, Row 0 is HTTP traffic, Row 1 is voice traffic and Row 2 is video traffic.
The next configuration is the WiMAX Base Station, here we have edited the WiMAX parameters to be similar to the WiMAX Config Settings and according to the Base Station Parameters in the Methodology. As we have done in the SS configurations, we have chosen Gold for all traffic as seen in Figure 49.

The last configurations are for the Applications Server, this is the server that SS nodes communicate with. The configurations of the server are seen in Figure 50, here we have configure all three application under the Application Supported Profiles and created a row for each application.
The figures above (Figure 49 and Figure 50) are the configurations of parameters of all the nodes in the WiMAX Network Model for the study.

4.3.3 Running the simulation

This stage is about running of the simulation, in this stage we choose what we want to simulate. The steps are as follows:

1. We first right click on the on an empty space of our simulation windows and select the “Choose Individual DES Statistics” as seen in Figure 51, this will display another windows
labelled Choose Results, in this window we expand Global Statistics and select what objects we want to simulate which is HTTP, Voice, Video Conferencing and WiMAX, see Figure 52.

After selecting objects to simulate press ok then run the run simulation button, see below Figure 53 the circled button. This button will display a simulation window (Figure 54) where we set the simulation duration, for this study the simulation duration is sixty (60) minutes, you can also choose 1 hour. Click the run button, this will display the progress of the simulation (Figure 55), when simulation is complete the window will display complete.
4.3.4 View and analyse results

In this stage we view the simulated results and do the analysis of the simulation. To view the simulated results we follow these steps:

1. Click on the View Results button (Figure 56) which will bring up a Results Browser window (Figure 57).
On the left bottom pane of the Results Browser, we expand the Global Statics to show us the simulated objects that we selected in step 1 of Section 4.3.3 (i.e. Running the Simulation), see Figure 58 for expanded view of these objects.
The main objective this study is to simulate the WiMAX performance, therefore it is important to note the sent and received traffic and also to note the WiMAX behaviour. To view these results we click on each checkbox.

To get the results for the HTTP simulation, sent and received traffic in bytes/second, we click under the HTTP and expand as shown in Figure 58 then check the boxes next to Traffic Received and Traffic Sent. The results are show in Figure 59.

To get the results for the Video Conferencing simulation, sent and received traffic in bytes/second, we click under the Video Conferencing (Highlighted in blue, on the left in Figure 58), expand as shown in Figure 58 then check the boxes next to Traffic Received and Traffic Sent. The results are show in Figure 60.

To get the results for the Voice simulation, sent and received traffic in bytes/second, we click under the, expand as shown in Figure 58 then check the boxes next to Traffic Received and Traffic Sent. The results are show in Figure 61.
The WiMAX Performance for the simulation of 20 SS when sending and receiving the traffic of the HTTP, Voice and Video Conferencing, can be obtained by checking the boxes under WIMAX (Shown in Figure 58). The boxes are the delay, load and throughput. The results for such are shown in Figure 62, Figure 63 and Figure 64 respectively.
The above images are for the simulation of the 20 Subscriber Stations, the next section will look into simulation of the 50 Subscriber Stations.

4.4 Simulation for Fifty (50) Subscriber Stations (SS)
The simulation of 50 SS follows the similar fashion as that of 20 SS described in section 4.3, in this section we will continually refer to the section 4.3. There is actually no difference in steps except for changing the number of SS to 50. In this section we will only show the images that differ to the previous section.

4.4.1 Creating Network Model
When creating the network model, we use the same method as in section 4.3.1. We followed step 1 to step 9, but in step 9 we specify the number of SS and we type 50 instead of 20 that was chosen previously.

From this step we move forward to other steps and do similar choices as in the simulation of 20 SS and the final Network Model for 50 SS is as seen section 4.3
4.4.2 Choosing Statistics and Running the simulation

The configuration for the 50 SS simulation and the running of the simulation follows the same steps as in the simulation of the 20 SS. We used the same configurations and the simulation window was also 60 minutes as in the simulation of 20 SS. The simulations for 75 SS and 100 SS will also have the similar steps for configuration of nodes and the running of the simulation will also be similar.

4.4.3 View and analyse results

The steps to view the results is also similar but the results were not the same. In Figure 67, Figure 68, Figure 69, Figure 70 and Figure 71 and Figure 72 are the results for the HTTP, video conferencing, voice and WiMAX.
Figure 68: Video Conference Sent and Received Traffic

Figure 69: Voice Sent and Received Traffic
Figure 70: WiMAX Delay in seconds

Figure 71: WiMAX Load in bits per second
As seen from above images, the performance of the WiMAX changed as we increased the number of SS also the sent and received traffic has slightly changed.

4.5 Simulation for Seventy Five (75) Subscriber Stations (SS)

In this section we will simulate 75 SS, we will follow the similar fashion as that of 20 SS and 50 SS, described in previous sections.

4.5.1 Creating Network Model

To create the network model, we will use the same method as that of 20 SS and 50 SS. We will follow same steps. The final network model for 75 SS is shown Figure 73 on the next page.
4.5.2 View and analyse results

The results for the simulation of the 75 SS are shown in the images below.
Figure 75: Video Conferencing Sent and Received Traffic

Figure 76: Voice Sent and Received Traffic
Figure 4.61: WiMAX Load in bits per seconds

Figure 77: WiMAX Delay in seconds

Figure 78: WiMAX Load in bits per second
4.6 Simulation for Hundred (100) Subscriber Stations (SS)

The last simulation of our study is the simulation of 100 SS, the WiMAX base station is designed to handle up to 100 SS. In this study we also want to see the performance of the WiMAX at 100 SS, we will follow the similar steps as that of 20 SS, 50 SS and 75 SS, described in previous sections.

4.6.1 Creating Network Model

The same method for creating the Network Model be used, we will therefore follow the same method as in section 4.3.1, all steps will followed but in step 9 we will change the number of SS to 100. The final network model for 100 SS is shown in Figure 80 on the next page.
4.6.2 View and analyse results

The simulation results for 100 SS are shown in the images below, note that the results are different for each number of SS.
Figure 82: Video Conferencing Sent and Received Traffic

Figure 83: Voice Sent and Received Traffic
Figure 84: WiMAX Delay in seconds

Figure 85: WiMAX Load in bits per second
After the simulation of the 100 SS we can see the performance of the WiMAX changes as you increase the number of SS, the sent and received data also changed with the addition of the SS. The next chapter will analyse the sent and received traffic also the performance of WiMAX for each simulation.
5 USE OF MARKOV MODEL TO ANALYSE PERFORMANCE

5.1 Introduction
Markov modelling is a modelling technique that is widely useful for dependability analysis of complex fault tolerant systems. It is very flexible in the type of systems and system behavior it can model. The Markov model is not limited to engineering field but is also used in other fields which include researches such as Speech Recognition, Bioinformatics, Offline Handwriting Recognition, Pharmacoeconomics and many other researches from diverse fields.

Markov is not, however, the most appropriate modelling technique for every modelling situation. The first task in obtaining a reliability or availability estimate for a system is selecting which modelling technique is most appropriate to the situation at hand. A person performing a dependability analysis must confront the question: is Markov modelling most appropriate to the system under consideration, or should another technique be used instead? [1].

In this study we have chosen the Markov modelling and deemed it as appropriate method for the analysis and simulation of our WiMAX Network. The Markov Modelling has been used by a number of researchers for the analysis of WiMAX performance, behavior and other related issues such as polling delay and throughput analysis. These previously conducted studies have been accepted and their result were satisfactory.

In this chapter we will first look at the similar work that has been done by other researchers as stated above, then look at the modelling of our WiMAX Network using Markov Chains and lastly at the simulation of the model.

In this study we will consider a single WiMAX cell with one (1) Base Station (BS) and twenty (20) Subscriber Station (SS) to hundred (100) SS and evaluate the performance of the WiMAX Network as we increase the number of SS. This has been the same scenario as in the previous chapter, Simulation using OPNET Modeler 14.5.

5.2 Similar Work: Markov Models for Network Simulations
Several studies related to the WiMAX performance analysis have been conducted in different environments and different methods of analysis have been tried and tested. In this section we are looking at studies related to the analysis of the WiMAX using Markov chains.

The researchers in [5] have done studies related to the performance Analysis of using Markov chains, in their paper they proposed a new analytical model for the performance analysis of various contention based bandwidth request mechanisms, including grouping and no-grouping schemes as suggested in the WiMAX standard. Their analytical model covers both unsaturated and saturated traffic load conditions in both error-free and error-prone wireless channels. The results of their
simulation and analysis show that the grouping mechanism outperforms the no-grouping mechanism when the system load is high, but it is not preferable when the system load is light and that the channel noise degrades the performance of both throughput and delay.

In [2] the study is about implementing the slotted ALOHA protocol as a two-state system, they have used the Markov Model to evaluate the channel utilization and have analyzed the throughput under different fairness conditions. In their conclusion they state that they got the results that are possible to achieve a desired system performance by suitably choosing the mean time channel occupancy. An aggregate throughput of at least one half can be achieved even if the number of competing users for the bandwidth increases. They have also noticed that the analysis of another performance which is the delay shows that the delay can be minimized.

The study in [6], the researchers on their paper analyze number of polls based on the Markov chain analysis to minimize the average polling delay while increasing network throughput. They proposed the analysis model of NPS in unicast polling for minimizing polling delay while increasing throughput and utilization. From the conclusion of their study it shows that the numerical results indicate that the analytic results of average polling delay and throughput are very close to the simulation results under moderate and heavy traffic load of packet transmission probability.

5.3 Modelling of WiMAX with Markov Chain

The simulation model for the study is explained in section 5.1 above, this model is reduced to Markov Model but is the same as the one simulated with OPNET Modeler 14.5 in Chapter 4. The radius that we have chosen to use is the same as that of previous chapter which is 64QAM (Quadrature Amplitude Modulation) as our Automatic Modulation Classifier (AMC), this was selected due to the understanding and fact that most of the rural towns in the former Transkei regions are about 6 kms in diameter when you put them in circular frame.

To control the medium access, MAC protocol coordinates the nodes in a network and resolves the contention among their accessing the shared medium so that the resources are shared fairly and efficiently [2]. The adaptive bandwidth request scheme selects the contention free scheme when the remaining bandwidth is enough to transmit at least one bandwidth request message. When contention-free-based option is selected, BS chooses a SS to transmit their bandwidth request based on the queue status. Queue status is the information of a packet in the queue of a SS. Using this approach, the collision probability can be reduced due to the decreasing number of contending SS. The proposal enhances the bandwidth efficiency and the data transmission delay of a SS is reduced as compared to the contention-based scheme [7]. The assumptions that we adopted are almost similar to those defined by Gebali in [8] where he models the analysis of IEEE 802.16 with Markov Chains. In this chapter we therefore construct a Markov Model from which we can analyse the throughput and delay of the WiMAX system, we will make the following assumptions:
The states of the Markov chain represent the number of queued users requesting access at the start of any time step.

- The Markov chain time step is taken equal to the frame duration.
- The system has a fixed population of \( m \) users.
- There is a single customer class.
- A user can have, at most, one message waiting for transmission. At the end of certain time step users that have requests pending cannot issue more requests at the next time step.
- Probability that a user requests transmission is \( P_0 \) and probability that the user is idle is \( 1 - P_0 \).
- All nodes are within the same transmission radius of the BS.
- All nodes have the same arrival rate and service rate.
- The BS uses the unicast polling method.

The unicast polling instead of group polling as mentioned in [6], is a useful method as it avoids the yielding contention delay due to bandwidth requests in group mode. Noting the above, we will use a two system state as follows: One state is the busy state when only one of the users is transmitting; the other state is either when the system is idle or when collisions happen [2].

### 5.3.1 Performance Analysis for WiMAX

In this section we analyse the performance of WiMAX, as explained in [8], to find the performance of the IEEE 802.16 system we need to obtain the transition matrix. We will model the performance of a WiMAX network through modelling the Access Point (AP) of a WiMAX since this is where all the user requests are processed and scheduled [8], we will use the Markov chain model to model our system. As stated above in section 5.3, we will consider a two system state, that is (1) the state is either busy state when only one of the Subscriber Station (SS) is transmitting, (2) the other state is either when the system is idle or when collision happens. We have therefore adopted the following notations for our Markov model:

- \( m \) = number of SS in the system
- \( P_t^i \) = Transmitting probability at free states for SS\( i \).
- \( P_r^i \) = Transmitting probability at backlogged states for SS\( i \).
- \( T_h^i \) = Throughput function, which indicates the average throughput of SS\( i \).

For all SS, we can assume that \( P_t^i = P_t \) and \( P_r^i = P_r \), using the formula in [2], the transition probabilities become:

\[
P_0 = mP_r (1 - P_r)^{m-1} \quad \text{indicating the probability that only one of the } m \text{ users is transmitting.}
\]

\[
P_c = (1 - P_r)^{m-1} \quad \text{indicating the probability that all of the } m-1 \text{ users is not transmitting}
\]

\[
P_0 = 1 - P_0 \]
The Transition matrix for the above explained Markov Chain is as follows:

\[
P = \begin{pmatrix} P_c & 1-P_c \\ P_0 & 1-P_0 \end{pmatrix}, \text{ where } P \text{ is a transition matrix.} \tag{10}
\]

The steady state distribution becomes the following:

\[
v = vP, \quad \text{with } v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \tag{11}
\]

where \(v\) represents the steady state distribution and whatever probability we have in a vector has to add up to 1, therefore:

\[
v_1 + v_2 = 1 \tag{12}
\]

Calculating \(v = vP\), we obtain the following:

\[
\begin{cases} 
v_1 \times P_c + v_2 \times P_0 = v_1 \\ v_1 \times (1-P_c) + v_2 \times (1-P_0) = v_2 \\ v_1 + v_2 = 1 \end{cases} \tag{13}
\]

We also find that

\[
v_1 = \frac{P_0}{1-P_c+P_0} \tag{14}
\]

and

\[
v_2 = \frac{1-P_c}{1-P_c+P_0} \tag{15}
\]
5.3.1.1 Retransmission Probability
In this type of system, [2] states that after one user successfully transmits a packet, it obtains the channel. This user will continue to occupy the channel for a random amount of time $T$. We will assume that $T$ is an exponential random variable with parameter $d = (1 - P_c)$. Let $E[T]$ the mean for the random variable $T$. If we put $E[T] = U$.

When $E[T] = d$, then

\[
E[T] = \frac{1}{d}
\]

Therefore, $U = \frac{1}{1 - P_c}$ \hfill (17)

\[
U = \frac{1}{1 - (1 - Pr)^{m-1}}
\]

(18)

5.3.1.2 Calculating the throughput of the system
In our calculation in 5.3.1, $v_1$ represents the total throughput of the system, therefore,

\[
v_1 = \frac{P_0}{1 - P_c + P_0}, \quad \text{where } v_1 = \text{total throughput}
\]

Using the calculations in 5.3.1.1 and the calculations in [2] by representing $v_1$ was a function of $U$ explained above in 5.3.1.1 we get the following:

\[
v_1 = \frac{mPr(1 - Pr)^{m-1}}{1 + (mPr - 1)(1 - Pr)^{m-1}}
\]

\[
= \frac{m(U - 1)}{m(U - 1) + 1/(1 - (1/U)^{m-1})}
\]

(20)
As seen on the above calculation, the throughput decreases when the transmission probability increases, when the probability becomes close to 1 the system throughput is almost zero. This validates the scenario shown in Figure 87 above.

5.3.1.3 Calculating the delay of the system

As stated in [2], to calculate the delay in the system we have to note the number of users in the system. Let the \( N \) be the number of users in the system, we get the following:

\[
N = v_1 x n_1 + v_2 x n_2
\]

Where \( n_1 = \sum_{i=1}^{m} i \) represents the number of users in the cases where the system can be in busy state.

And \( n_2 = \sum_{i=2}^{m} i \) represents the number of users in the cases where a collision is probable (the system can be in an idle or collision state).

\[
N = v_1 \times \left[ \frac{m \times (m+1)}{2} \right] + v_2 \times \left[ \frac{(m-1) \times (m+2)}{2} \right]
\]  

(22)

by using the formula of LITTLE [9], the delay is calculated as

\[
D = \frac{N}{v_1}
\]

(23)

therefore the delay can be calculated as the following:

\[
D = \frac{m(m+1)}{2} \times \frac{(m-1)(m+2)}{2m(1-n^{-\frac{1}{u}}(1-1/u))(u-1)}
\]

(24)

5.4 Simulation with MATLAB R2011a

The Markov Model of a WiMAX system explain in the above sections will be simulated using MATLAB R2011a. In this section we will look at the simulation of a throughput and the delay of a WiMAX System using the equations explained in the section above. Here will first briefly explain what MATLAB is then we will show images on how we used the formula in MATLAB for calculating the throughput and delay at different values of Retransmission Probability (\( U \)). The values we have chosen for \( RB \) is 2, 8, 16 and 24. In MATLAB we are able to plot multiple graphs in one X-Y plane, we will therefore plot a throughput graph with number of SS on our X-axis and throughput resulting values on Y-axis at different values of \( U \).
5.4.1 MATLAB
MATLAB is a programming language created with the basic goal of providing a simple intuitive platform for engineering design applications, it can also be classified as a software for computation and visualisation [10]. It is an interactive working environment in which the user can carry out quite complex computational tasks with few commands. It was originally developed in 1970s by Cleve Muller. The results of MATLAB calculations can be view both numerically as well as in the form of 2D as well as 3D graphs easily and quickly [11].

5.4.2 Throughput Simulation
In this section we are simulating the throughput of the Markov Model at different values of $U$. These values have not made a significant change in the performance of the system, you will note that at the higher values of $U$ system throughput does not change as you increase the number of SS.

![Figure 87: Throughput vs Number of SS at different values of $U$](image)

Figure 87 below shows the MATLAB Desktop, the middle top window is where we write a script to calculate the throughput using the formulas in Section 5.3.1.2. The bottom middle window is where we run the scripts. Note that we have initialised different $U$ integer values and throughput integer values but the number of SS will start from one (1) to hundred (100) using a `while` loop. The output is them an array of SS in the X-axis and the throughput in the Y-axis as seen in Figure 88, note that that the throughput values are almost similar as the values of $U$ increases.
Below is the code for the throughput calculation:

```matlab
close all;
h1 = figure(1);
m=1; U=2;
U8=8;
U16=16;
U24=24;
M=m;

th=[m*(U-1)/(m*(U-1)+ 1/(1-(1-1/U)^p))]; th8=[m*(U8-1)/(m*(U8-1)+ 1/(1-(1-1/U8)^p))]; th16=[m*(U16-1)/(m*(U16-1)+ 1/(1-(1-1/U16)^p))]; th24=[m*(U24-1)/(m*(U24-1)+ 1/(1-(1-1/U24)^p))]; while m < 100
p=m-1;

Th=m*(U-1)/(m*(U-1)+ 1/(1-(1-1/U)^p)); Th8=[m*(U8-1)/(m*(U8-1)+ 1/(1-(1-1/U8)^p))]; Th16=[m*(U16-1)/(m*(U16-1)+ 1/(1-(1-1/U16)^p))]; Th24=[m*(U24-1)/(m*(U24-1)+ 1/(1-(1-1/U24)^p))]; th=[th,Th];
th8=[th8,Th8];
th16=[th16,Th16];
th24=[th24,Th24];
m=m+1;

%U=m;
M=[M,m]; end

plot(M,th,'-k'); hold all; plot(M,th8,'-r'); hold all;
```

Figure 88: MATLAB R2011a Desktop
plot(M,th16,'-b'); hold
all plot(M,th24,'-m');
hold off;
title('Throughput vs Number of Subscriber Stations at different levels of U')
YLabel('Throughput')
XLabel('Number of Subscriber Stations')
text(70,0.6,'U=2:   Black','FontSize',10)
text(70,0.55,'U=8:   Red','FontSize',10)
text(70,0.45,'U=24: Magenta','FontSize',10)

5.4.3 Delay equation in MATLAB
The way in which we calculated the delay of the WiMAX System is similar to that of throughput,
we created a script in for the delay in MATLAB script area. The delay output decreases as we
increase U, also the delay increases as we increase the number of SS. Higher values of U tend to
produce similar values of delay. In Figure 90, is the output graph for the delay vs the number of
SS at different values of U.

![Figure 89: Delay vs Number of SS at different values of U](image)

Below is the code used for calculating the delay vs Number of SS at different values of U:

close all;
h2 = figure(2);
m=10;
M=m;
u=2;
u8=8;
u16=1
6;
u24=24;

a=m*(m+1); 
b=(m-1)*(m+2); 
d=2*m; 
h=1-(1/u); 
h8=1-(1/u8); 
h16=1-(1/u16); 
h24=1-(1/u24); 
i=1/(m-1); 
g=h^i; g8=h8^i; 
g16=h16^i; 
g24=h24^i; 
e=1-g; 
e8=1-g8; 
e16=1-g16; 
e24=1-g24; 
f=u-1; f8=u8-1; 
f16=u16-1; 
f24=u24-1; 
c=d*e*f; 
c8=d*e8*f8; 
c16=d*e16*f16; 
c24=d*e24*f24; 
Dly = a/2 + (b/c); 
Dly8 = a/2 + (b/c8); 
Dly16 = a/2 + (b/c16); 
Dly24 = a/2 + (b/c24); 

while m < 100

a=m*(m+1); b=(m-1)*(m+2); d=2*m; 
h=1-(1/u); 
i=1/(m-1); 
g=h^i; 
e=1-g; f=u-1; 
c=d*e*f; 
D = a/2 + (b/c); D8 = a/2 + (b/c8); D16 = a/2 + (b/c16); D24 = a/2 +
The results of both the delay and the throughput show the reflection of the system performance at different stages retransmission probability. The Markov Modelling cases which we have come across do not reflect the time factor in their calculation and therefore we have followed suite. In our simulation using OPNET we have throughput and delay both against time at different number of SS. In MATLAB simulations we have Throughput and delay against number of SS at different values of retransmission probability. But we have noted that the delay increased as the number of SS increase for OPNET and Markov chain modelling.

The same is true for the throughput, as we increase the number of retransmission probability they tend to have similar results at any number of SS, such that as you increase the number of SS the throughput slightly increases in value. This is also the same with OPNET simulation the throughput
is 8Mbits/sec, 10Mb/s, 10.5Mbit/sec and 11Mbits/sec for 20 SS, 50 SS, 75 SS and 100 SS respectively. If these results were plotted in MATLAB using throughput vs SS, the graph would be similar to that of the results achieved in this section of the study.
6 SIMULATION USING NS-3.19

6.1 Introduction
In this chapter we will simulate our WiMAX Network Performance using NS-3.19, this version of NS-3 was chosen from all other versions because it has all the required packages for simulating WiMAX and has been tested. The latest versions of NS-3 could not satisfactorily run some of the practice tutorials as they returned errors. NS-3.19 returned no errors and was able to plot graphs using gnuplot and the animated simulation using NetAmin returned satisfactory results.

The simulation platform for NS-3.19 is not the same as that of OPNET and MATLAB, these latter two simulator are Windows-based programs but NS-3.19 is an open source program. The platform at which NS-3.19 was implemented is the following:

- Manufacturer: Hewlett-Packard
- Model: HP EliteBook 2570p
- Processor: Intel Core i5, 2.50GHz
- Installed Memory: 4.00 GB
- System Type: Ubuntu 15.04 - 64bit version.

We will follow the steps defined in the previous chapter which is Methodology which outlined the following steps:

- Converting the nodes of a network model to a C++ program.
- Running the program in Command Line Interface (CLI) using waf command of NS-3
- Run Simulation
- Plotting Graph of the simulation output in CLI Gnuplot [115]
- Animate using NetAmin [116]
- Analyse the output graph

The strategy that we will adopt is that we used when we simulated using OPNET, at first we simulated a scenario with Twenty (20) Subscriber Stations (SS) nodes, then Fifty (50) SS, Seventy Five (75) SS and lastly Hundred (100) SS.

All our simulations topology will at list consist of the following:

- WiMAX Base Station
- WiMAX Mobile Workstation
- Links between hosts and devices
- Applications Server
6.2 Simulation of Twenty Subscriber Stations.
In this section we will simulate Twenty (20) Subscriber Stations (SS), we will translate our WiMAX Network Topology of 20 SS into C++ language, then simulate using the steps explained in the previous section, section 6.1.

6.2.1 Converting the nodes of a network model to a C++ program.
In this step we convert the explained topology into C++ language, this is the program language used in all NS-3 versions. Because of the length of the program, we will only explain the important part of the program. Here we will show the 20 SS portion, and all our SS and the Base Station are defined as nodes, we will therefore show a part that explains the configuration of all nodes including the portion that shows the server (Voice, Video and HTTP). This script is written using the Linux text editor called the gedit. This is similar to the notepad in Windows. We have named this script WiMAX20.cc.

- Defining Base Station (BS) and Subscriber Stations (SS)
  ```
  int nbSS = 20; //num_of_SS value
  
  ssNodes.Create (nbSS);
  bsNodes.Create (1);
  ```
Here we initialise the BS and the SS's, the value of SS is initialised to 20 and the BS is set to 1. The program has been informed that it will contain one BS and twenty SS. This is the same as in OPNET Topology configuration where we were defining the number of SS and the number of BS.

Defining Voice Traffic
```
uint16_t VoIPPort = 84;

//creation of VoIP traffic
Address VoIPsinkAddress (InetSocketAddress (StreamerCSMAInterfaces.GetAddress(0), VoIPPort));
PacketSinkHelper VoIPpacketSinkHelper ("ns3::UdpSocketFactory", InetSocketAddress (Ipv4Address::GetAny (), VoIPPort));
ApplicationContainer VoIPsinkApps = VoIPpacketSinkHelper.Install (ssNodes.Get(0));
VoIPsinkApps.Start (Seconds (0));
VoIPsinkApps.Stop (Seconds (duration-1));
```

// Classifier for VoIP Traffic

---

86
IpcsClassifierRecord DlClassifierUGS (SSinterfaces.GetAddress (4),
    Ipv4Mask ("255.255.255.255"),
    ASNCSMAInterfaces1.GetAddress (0),
    Ipv4Mask ("255.255.255.255"),
    0,
    65000,
    VoIPPort,
    VoIPPort,
    17, //IRTP
    3);
ServiceFlow DlServiceFlowUGS = WiMAX.CreateServiceFlow (ServiceFlow::SF_DIRECTION_UP,
    ServiceFlow::SF_TYPE_UGS,
    DlClassifierUGS);
ss[4]->AddServiceFlow (DlServiceFlowUGS);

Here we initialise the Voice traffic to operate in the IPv4 network, we set the data rates, the source
and interface for Voice Traffic. We recommended that WiMAX be run on the IPv4 network as this
is the current and widely used and tested version. The IPv6 protocol is currently not as popular as
IPv4, IPv4 is the one that widely accepted and where most devices operate.

Defining Video traffic
uint16_t VideoPort = 83;
.
//creation of Video traffic
Address VideosinkAddress (InetSocketAddress (SSinterfaces.GetAddress(3), VideoPort)); // interface of Application server
PacketSinkHelper VideopacketSinkHelper (*ss3::UdpSocketFactory", InetSocketAddress (Ipv4Address::GetAny (),
    VideoPort));
ApplicationContainer VideosinkApps = VideopacketSinkHelper.Install (ssNodes.Get(3)); //Application Sink
VideosinkApps.Start (Seconds (0));
VideosinkApps.Stop (Seconds (duration-1));

Ptr<Socket> Videos3UdpSocket = Socket::CreateSocket (Streamer_Node.Get (0), UdpSocketFactory::GetTypeId ()); //source

Ptr<MyApp> Videoapp = CreateObject<MyApp> ();
Videoapp->Setup (Videos3UdpSocket, VideosinkAddress, 1024, 1000, DataRate ("32kbps"));
Streamer_Node.Get (0)->AddApplication (Videoapp);
Videoapp->SetStartTime (Seconds (1.));
Videoapp->SetStopTime (Seconds (duration-2));

Here we have initialised the Video Traffic, its data rate, port also the interface.

Defining HTTP Traffic
uint16_t HTTPPort = 80;
.
.

// creation of HTTP traffic
Address HTTPsinkAddress (InetSocketAddress (SSinterfaces.GetAddress(0), HTTPPort)); // interface of Application server
PacketSinkHelper HTTPpacketSinkHelper ("ns3::TcpSocketFactory", InetSocketAddress (Ipv4Address::GetAny (), HTTPPort));
ApplicationContainer HTTPsinkApps = HTTPpacketSinkHelper.Install (ssNodes.Get(0)); // Application Sink
HTTPsinkApps.Start (Seconds (0));
HTTPsinkApps.Stop (Seconds (duration-1));
.
.
// Classifier for HTTP Traffic
IpcsClassifierRecord DlClassifierBe (StreamerCSMAInterfaces.GetAddress (0),
 Ipv4Mask ("255.255.255.255"),
SSinterfaces.GetAddress (0),
Ipv4Mask ("255.255.255.255"),
HTTPPort,
HTTPPort,
0,
65000,
6,
0);
ServiceFlow DlServiceFlowBe = WiMAX.CreateServiceFlow (ServiceFlow::SF_DIRECTION_DOWN,
  ServiceFlow::SF_TYPE_BE,
  DlClassifierBe);
ss[0]->AddServiceFlow (DlServiceFlowBe);

Here we have initialised the HTTP Traffic, its data rate, the port and the flow of traffic.

6.2.2 Running program in Command Line Interface (CLI) using waf command of NS-3

For the program to run properly without errors we need to make sure that the Linux packages for the C++ are installed. To run the script, the first thing to do is to open the Terminal in Linux computer (by default it takes you to the home folder as shown in Figure 92) and enter in to the folder of the ns-3.19 (by using cd command, shown in Figure 93). In our case we have extracted the NS-3 in the /home/sandla/ns-allinone-3.19 and all files related to NS-3, shown in Figure 94. The script is saved in the scratch folder, when in terminal this file is in /home/sandla/ns-allinone-3.19/ns-3.19/scratch. To access the Terminal, you click on the top icon in the far right of the ubuntu default menu see Figure 91, this requires basic knowledge of ubuntu desktop. When we open the Terminal program it will take us into home folder as shown below.
After clicking on the terminal icon, it shows a terminal window as shown below in Figure 92.

As stated above, below is how you access the main containing folder for ns-3.19 program using the command line in the terminal program, see Figure 93.

In the main ns-3.19 folder is where you run all programs related to ns-3, all files for installation and running of the program are found in the main folder as shown in Figure 94.
The important folder for ns-3 is the ns-3.19 folder, this where we run scripts and scripts are saved within this folder, Figure 94 shows the content of the folder. To get to the ns-3 folder we type the cd command the point to the ns-3.19 folder as show in Figure 95.

Figure 94: ns-3.19 folder

In Figure 95, we have `sandla@nkaules:~/ns-allinone-3.19$ cd ./ns-3.19` then we press enter to get into the folder to get into `/home/sandla/ns-allinone-3.19/ns3-19` folder.

Figure 95: list of files and folders inside ns-3.19 folder.

The ns-3.19 folder contains the waf program to run C++ scripts, it is recommended to save the scripts in the scratch folder which is inside [ns-3.19] this folder. Figure 96 shows the list of files and folders inside the ns-3.19.

After finishing the WiMAX20.cc script, we saved it in the scratch folder. Figure 97 shows command to get into the scratch and Figure 98 shows the list of files inside the scratch folder.

Figure 96: Command to access scratch folder
To run the script, we need to be on the ns-3.19 folder and then type the following command:

```
./waf --run scratch/WiMAX20 --vis
```

then press enter to run as show in Figure 99 below.

Figure 97: List of files in scratch folder

6.2.3 Run Simulation

In Figure 100 we are typing a command for running the simulation of the 20 SS, the script WiMAX20.cc is the script for 20 SS. Note that the .cc is omitted when we run the script using the ./waf command.

The output of the simulation is shown in Figure 101 and Figure 102, this also brings out the animation window where we can observe the transfer of packets from the applications server to the 20 SS. This animation window allows us to re-run the simulation and also we can increase or decrease the speed of simulation.

Figure 98: Running simulation using the ./waf command for 20 SS

Figure 99: ./waf program simulating the WiMAX20.cc file.
The program has detected 23 nodes, this 23 is made of 20 SS and one BS, the other two nodes is the internet router node and the other node represents the applications server. On the animation window when we press the simulate button or F3 on the keyboard it run the animation of the simulation (see Figure 101) and at the same time prints the output on the terminal as seen in Figure 103. When the Python animation is complete it shows all nodes and the traffic flow, see Figure 102.

![Figure 100: Animation with Python before pressing Simulation Bottom.](image)

![Figure 101: Animation with Python after pressing Simulation Bottom.](image)

In Figure 103 the output of the simulation is show in the terminal, it shows us the transmitted packets, received packets, lost packets, total throughput of the system and the delay in the system.
6.2.4 Plotting Graph of the simulation output in CLI Gnuplot

Inside our script we have defined the plotting of throughput graph and the delay graph for Gnuplot, the lines can be found in our script WiMAX20.cc, the following are the lines for managing the plotting of the throughput and delay graphs.

```cpp
std::ofstream ofs("Throughput20.plt", std::ofstream::out);
std::ofstream ofs1("Delay20.plt", std::ofstream::out);

ofs << "set terminal png" << std::endl;
ofs << "set output 'Throughput20.png'" << std::endl;
ofs << "set title 'WiMAX'" << std::endl;
ofs << "set xlabel ''" << std::endl;
ofs << "set ylabel 'value'" << std::endl;
ofs << "plot '-' title 'Throughput_HTTP' with linespoints, '-' title 'Throughput_video' with lines", '-' title 'Throughput_voip' with lines" << std::endl;

ofs1 << "set terminal png" << std::endl;
ofs1 << "set output 'Delay20.png'" << std::endl;
ofs1 << "set title 'WiMAX'" << std::endl;
```

---

**Figure 102:** Output of the simulation using `./waf` command in Terminal.
ofs1 << "set xlabel ''" <<std::endl;
ofs1 << "set ylabel 'value(s)''" <<std::endl;
ofs1 << "plot " <<"'-'" <<"title " <<"'delay_HTTP' with linespoints, '-' title 'delay_video' with lines, '-' title 'delay_voip' with lines" <<std::endl;

To access the Gnuplot workspace we need to type the command `gnuplot` in the `ns-3.19` folder, as seen in Figure 104 below.

![Figure 103: gnuplot command and gnuplot workspace.](image)

These following lines below (which we wrote in our script `WiMAX20.cc`):

```cpp
std::ofstream ofs ("Throughput20.plt", std::ofstream::out);
std::ofstream ofs1 ("Delay20.plt", std::ofstream::out);
```

They are for calling the files `Throughput20.plt` and `Delay20.plt`, these files are run in the gnuplot workspace and they plot the graphs and save them in `.png` format.

To call these files to generate the graphs we run load command and specify the file you want to run, to generate the delay graph we run the `load 'Delay20.plt'` file and for the throughput graph we run the `load 'Throughput20.plt'` as seen in Figure 105 and Figure 106. The output graphs are the delay and the throughput as seen below in Figure 107 and Figure 108.
Figure 104: gnuplot command for generating a Throughput graph

Figure 105: gnuplot command for generating a Delay graph
Figure 106: Throughput of the system at 20 SS.
6.2.5 Animate using NetAnim

The ns-3.19 also allows us to simulate using the NetAmin program, this program can be found inside the netanim folder which is the list of folders listed under the ns-allinone-3.19 folder, see again in Figure 94. The content of the netanim folder is shown in Figure 109 below.

To run the NetAnim program you need to access the netanim-3.104 and run the NetAnim folder, this will open a NetAnim window as seen in Figure 110 below.
When the NetAnim window shows up, we then click on the open icon, this will allow us to select the xml, in our case we will select the WiMAX20.xml as seen below in Figure 111.

In Figure 111, we have the window open to select WiMAX20.cc, we then click the open button and it will display a window with animation space shown in Figure 112 below.
When we run the program after selecting a file, we click the play button and we will get the animation and this is similar to the one we saw in Python animation. The animations snapshots can be seen in Figure 113 below.

6.3 Simulation of Fifty Subscriber Stations.
In this section we will simulate Fifty (50) Subscriber Stations (SS), we will first translate our WiMAX Network Topology of 50 SS into C++ language as we did in section 6.2, then simulate using the steps explained section 6.1. The steps of simulation are the same as those of simulating the 20SS, we will therefore skip some of the similar sub-steps.
6.3.1 Converting the nodes of a network model to a C++ program.
This section is the same as that of Simulation of 20 SS, the difference here is that the number of SS nodes is 50 as we will see below. The script for simulating 50 SS is *WiMAX50.cc*, here we have initialised the nbSS to 50 as seen in the step below.

Defining Base Station (BS) and Subscriber Stations (SS)

```c++
int nbSS = 50;
.
.
ssNodes.Create (nbSS);
bsNodes.Create (1);
```

The traffic is the same as that defined in the simulation of the 20 SS, as seen in *WiMAX20.cc* script. We will therefore skip the section of defining the type of traffic.

6.3.2 Running program in Command Line Interface (CLI) using waf command of NS-3
The simulation steps of the 50 SS are the same as those of 20 SS, we use the similar approach and the files that need to be installed are already installed and will be the same. The folders will also be the same but the difference will be the will be name of the script, here we will run the *WiMAX50.cc* in the terminal. In the terminal we will therefore run the following command:

```
./waf --run scratch/WiMAX50 --vis
```

In our scratch we have placed the *WiMAX50.cc*, and in Figure 114 we run the script using the
```
./waf --run scratch/WiMAX50 --vis
```

![Figure 113: Running simulation using the ./waf command for 50SS.](image)
6.3.3 Run Simulation

Running the simulation of 50SS using the *waf* command produces an output shown in Figure 115 below.

![Figure 114: Animation with Python before pressing Simulation button for 50SS](image1)

This window is automatically displayed after running the *waf* command and it stays the same until the Simulate button is pressed, when Simulate button is pressed it produces output in terminal and at the same time run the animation Python window. Both Figure 116 and Figure 117 are screenshots of the output after pressing the Simulate button.

![Figure 115: Animation with Python and the output in terminal after pressing Simulation button.](image2)
6.3.4 Plotting Graph of the simulation output in CLI Gnuplot

Plotting the graph using gnuplot command requires that we run the `gnuplot` in terminal and then we load the `.plt` files. These files are defined in WiMAX50.cc script.

![gnuplot command and gnuplot workspace.](image)
Figure 118: Throughput of the system at 50 SS.

Above in Figure 118 is the gnuplot workspace and the command to generate the throughput and the delay for the 50 SS. Figure 119 is the throughput output graph and Figure 120 is the delay output graph as generated in Figure 118.
6.3.5 Animate using NetAnim

The steps for running NetAnim are the same as that of 20SS, but for 50SS we select a WiMAX50.xml file. The outputs for NetAnim are shown in Figure 121 and Figure 122 below.

Figure 120: NetAnim output with packets from Application server to BS.

Figure 119: Delay of the system for at 50 SS.
6.4 Simulation of Seventy Five Subscriber Stations.
In this section we will simulate Seventy Five (75) Subscriber Stations (SS), we will translate our WiMAX Network Topology of 75 SS into C++ language, then simulate using the steps explained in the previous section, section 6.1. Simulation steps are similar to the previous simulations, we therefore will omit some of the steps and dwell more on points of difference.

6.4.1 Converting the nodes of a network model to a C++ program.
The conversion is the same as in the previous simulations but the difference is in the number of Subscriber Stations, which is Seventy Five (75) in this case. The script for simulating 75 SS is *WiMAX75.cc*, here we have initialised the nbSS to 75 as seen in the step below.

- Defining Base Station (BS) and Subscriber Stations (SS)

```c++
int nbSS = 75;
```

The traffic settings are the same as that defined in the previous simulations, we will therefore skip the section of defining the type of traffic.
6.4.2 Running program in Command Line Interface (CLI) using waf command of NS-3

The simulation of 75 SS uses the same method as for previous simulations also the folder where this file is saved is the same folder. The difference here is that the script for 75SS is WiMAX75.cc, and when running in the terminal we type the following command:

```
./waf --run scratch/WiMAX75 --vis
```

Below in Figure 123, we have an image of how we run the script using the `waf` command.

![Figure 122: Running simulation using the ./waf command for 75SS.](image)

6.4.3 Run Simulation

When running the simulation we press enter after typing the command shown in Figure 123 and this will run the simulation in terminal as shown in Figure 125 this will at the same time run the Python animation program to visualise the running of the simulation in the terminal, Figure 124 and Figure 126 show us the simulation in terminal running in parallel to Python animation.

![Figure 123: Running simulation in terminal and Python before pressing the Simulate button.](image)
Figure 124: Running simulation in terminal and Python after pressing the Simulate button.

Figure 125: Simulation running in terminal and Python.
6.4.4 Plotting Graph of the simulation output in CLI Gnuplot

The plotting of graph using gnuplot command we will run the `gnuplot` in terminal and then run the load command in Gnuplot workspace, we need to run the `.plt` files. These files are defined in WiMAX75.cc script and are automatically produced at the program was running in section 6.4.3 and these are automatically stored in `ns-3.19` folder.
Below is the images showing the loading of .plt files in Gnuplot workspace, to access the Gnuplot workspace we use the same command as shown in Figure 129 and the explanation is the same as in section 6.2.4 above but the difference is that we use WiMAX75.cc script, more of WiMAX75.cc, the Figure 129 shows the loading of files in Gnuplot workspace.

![GNUPLOT Image](image_url)

**Figure 128: Loading plot files for 75 SS.**

The output file of these will be plotted graphs in .png format, as mentioned earlier will be found in the ns-3.19 folder we have named them Delay75.png and Throughput75.png file, see below in Figure 130 and Figure 131 for these output files.
Figure 129: Delay of the WiMAX System at 75 SS.
6.4.5 Animation using NetAnim

The steps for running NetAnim are the same as that of 20SS in section 6.2.5 above, for 75SS we select a WiMAX75.xml file this file was created during the simulation of WiMAX75.cc script. The screenshots of the NetAnim are shown in Figure 132 and Figure 133 below, note that this now has 75 SS nodes and one node for BS and the other node is for router the last node is the Application Server, so we therefore have 78 nodes in total. This number of nodes if you have noted, is the same as that of the simulation in OPNET. Note also this number is shown when running the script in terminal using the waf command.
6.5 Simulation of Hundred Subscriber Stations.

Here we will simulate Hundred (100) SS and the steps are still similar to the previous simulation but here we use WiMAX100.cc, here we defined inside the script that the number of SS should be 100. Also note at the start of simulation that the output in terminal finds three (3) additional nodes, this is because the programs recognises the initialised number of SS and also adding to that the BS, the IP Gateway or Router and lastly the Applications Server. It is therefore expected that the nodes that will recognised by the program is 103 nodes.
6.5.1 Converting the nodes of a network model to a C++ program.
The C++ script for simulating 100 SS is WiMAX100.cc, inside the script we have initialised the
nbSS, which the number of SS to 100 as seen in the step below.

Defining Base Station (BS) and Subscriber Stations (SS)

```
int nbSS = 100;
```

The traffic settings are the same as that defined in the previous simulations, we will therefore skip
the section of defining the type of traffic.

6.5.2 Running program in Command Line Interface (CLI) using waf
command of NS-3

The simulation of 100 SS will require the script WiMAX100.cc the command to run the script is
seen below and note that the .cc is not part of the command and is written as seen below.

```
./waf --run scratch/WiMAX100 --vis
```

Below in Figure 134, we have an image of how we run the script using the waf command.

![Figure 133: Running simulation using the ./waf for 100 SS](image)

6.5.3 Run Simulation

Execution of the .waf command for WiMAX100.cc script produces an output shown in Figure 135,
note that after executing the .waf command the window is displayed and the programs requires
pressing of the Simulate button for the program to continue with the simulation. In Figure 136,
Figure 137 and Figure 138 we see that the program is running and starting to produce output in
both Python and terminal.
Figure 134: Terminal and Python output before pressing Simulate button.

Figure 135: Terminal and Python output after pressing Simulate button.
6.5.4 Plotting Graph of the simulation output in CLI Gnuplot

The gnuplot command is run in the terminal and then inside the Gnuplot workspace we run the load command, this loads the .plt files. These files are defined in WiMAX100.cc script and are automatically produced when we were running the .waf in section 6.5.3 and these are automatically stored in ns-3.19 folder.

These files loaded in Gnuplot program using the load command as see in Figure 139, this then creates the .png graphs that plot the throughput and delay. In Figure 140 and Figure 141 the output plots for the delay and throughput respectively.
Figure 138: gnuplot command and gnuplot workspace, loading .plt files.

Figure 139: Delay for the WiMAX System at 100 SS.
6.5.5 Animate using NetAnim

To run the NetAnim for animation of 100 SS, we select the \texttt{WiMAX100.xml} file which was generated during the simulation using \texttt{waf}. To run NetAnim we type the command in terminal to get a NetAnim window and we select the \texttt{WiMAX100.xml} file and press the run button to start simulation as seen in the previous NetAnim simulations. The screenshots of the NetAnim are shown below in Figure 142 and Figure 143 and note that the number of nodes is 103 in total. This is because we have 100 nodes and added to that is one node for BS and the other node is for router the last node is the Application Server. This number is also shown in the simulation using the \texttt{waf} command.
The NetAnim simulator shows SS in red colour, the BS is in green colour, the Router/Gateway is in yellow colour and the Applications Server is in grey colour.
7 DISCUSSION OF RESULTS

The simulation of WiMAX performance was conducted and as seen above the results of simulation for each simulator were produced. The simulation was conducted using the OPNET Modeler in Windows 7, Network Simulator in Ubuntu 15.04 Linux and Markov Chains which we simulated using MATLAB in Windows 7.

7.1 Comparison of Results
Table 4 and Table 5 below display the output results of delay and throughput respectively for the mentioned simulators.

Table 4: Simulation Results - Delay

<table>
<thead>
<tr>
<th>No. of SS</th>
<th>OPNET</th>
<th>Markov(MATLAB)</th>
<th>NS-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>17 sec</td>
<td>0.7 sec</td>
<td>1 ~ 5sec</td>
</tr>
<tr>
<td>50</td>
<td>25 sec</td>
<td>3 sec</td>
<td>1 ~ 5sec</td>
</tr>
<tr>
<td>75</td>
<td>27 sec</td>
<td>8 sec</td>
<td>1 ~ 5sec</td>
</tr>
<tr>
<td>100</td>
<td>27.5 sec</td>
<td>12 sec</td>
<td>1 ~ 5sec</td>
</tr>
</tbody>
</table>

Table 5: Simulation Results - Throughput

<table>
<thead>
<tr>
<th>No. of SS</th>
<th>OPNET</th>
<th>Markov(MATLAB)</th>
<th>NS-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>8 Mbits/sec</td>
<td>1Mbits/sec</td>
<td>140 kbits/sec</td>
</tr>
<tr>
<td>50</td>
<td>9.8 Mbits/sec</td>
<td>1Mbits/sec</td>
<td>140 kbits/sec</td>
</tr>
<tr>
<td>75</td>
<td>10.5 Mbits/sec</td>
<td>1Mbits/sec</td>
<td>140 kbits/sec</td>
</tr>
<tr>
<td>100</td>
<td>11 Mbits/sec</td>
<td>1Mbits/sec</td>
<td>140 kbits/sec</td>
</tr>
</tbody>
</table>

7.2 Comparison with previous studies

The previous studies related to the simulation of the WiMAX system performance have been conducted, in this section we look at few of the studies that these experts have conducted and compare with our own study and to see if our study agrees with theirs.

7.2.1 Simulation of WiMAX System Performance using OPNET

In this section we are comparing our OPNET simulation results with previous OPNET studies that were conducted. In [108] the study is done by an assistant professor and the MTech scholar. Here
they used OPNET modeller 14.5 to simulate the performance analysis of Physical layer of WiMAX Network, in their study they have three (3) cells with 3 SS each, these cells connect to the Internet backbone, their network topology is similar to our topology except that we have one cell and with 20 SS at first then increment to 100. The trend of the output graphs of their simulations show relatively similar output of our results in terms of amount of delay and throughput. The delay increases and the throughput decreases as the time goes by. The researchers have confirmed that the less the number SS the better the performance of the system and this is confirmed in the output graphs. The situation is the same for our study, where we have 20 SS we have better performance compared to 100 SS when using OPNET Modeler 14.5. Also note that we have 64 QAM which has better performance and therefore the results are relatively comparable to theirs in terms of performance. This is also in-line with the study conducted in [109] where they use the modeler to study networks with 15, 25 and 40 mobile workstations. In each network, group of five WiMAX workstations connect and call each other through one WiMAX base stations during 1000sec. Here the performance of parameters that indicate the quality of services such as initial ranging activity, delay, total transmission power and PHY path loss have been studied. In their conclusion they stated that Queuing delay is highest in large geographic area or with large no. of users and lowest in small users’ area. WiMAX PHY Path loss and Total transmission power of base stations is lowest in medium network and highest in small network that is with lesser no. of nodes. Neighbor advertisements sent are maximum in densely populated area and minimum in less populated area. The same is true for our study, the performance is better when we have less number of nodes.

7.2.2 Simulation of WiMAX System Performance using NS-3

The NS-3 is an open source network simulator which is made from C++, the scripts for simulation are written in C++ and are dependent on classes and functions provided by NS-3. The output results of the NS-3 simulator provided in chapter 6 and the animation was also provided. The network topology is the same as that of OPNET modeller 14.5 and the animated output displayed similar topology as OPNET. The animation was done by Phyton and NetAnim, the output graphs are created by GnuPlot. The Gnuplot output graph compared to the other NS-3 studies done previously provide relatively similar output. The study in [110] shows similar trend with our research related to NS-3. The output of the graph shows almost constant value of throughput and delay, the difference in our case was that the delay increased at the time increase but the throughput was almost a constant value as it is with their study. The study in [111] has used similar C++ functions for calculating delay and throughput. Browsing through the internet we have found a number of examples that calculate the performance in similar manner, the HTTPS://groups.google.com/forum/#!forum/ns-3-users is dedicated to NS-3 research and we have used the recommended solutions for calculation of WiMAX performance.

7.2.3 Simulation of WiMAX System Performance using Markov Chain

In similar studies related to Markov Chain where they calculate the WiMAX performance, the studies have opted to calculate using a two system state, the example of that is in [112] where they calculated the performance of the WiMAX system by looking at the fact that the BS system
communicates with one SS at a time and that creates a two state system at any time when transmission takes place in the system, this method is used by other studies such as in [113] and in [64].

8 CONCLUSION AND FUTURE WORK

The purpose of this chapter is to discuss the summary and the conclusion of the results obtained on this research and also to propose areas of future research.

8.1 Conclusion

In this research we do simulation of performance of WiMAX using simulators and Markov chains. The outcome of the simulation gave satisfactory outputs in different ways, OPNET has been noted as the one that produced better outcomes compared to the other methods of simulations used in this study. The OPNET simulation software was more user friendly and easy to understand when it comes to building the network model compared to the NS-3 and Markov Chains. Then NS-3 requires C++ knowledge and understanding of Linux operating system, the program can work only under the Unix/Linux operating systems. Unlike OPNET, NS-3 is free of charge and has an easily accessible knowledge hub and online support. The NS-3 animation output has displayed a true representation of the system in our study according to our specified model. The Markov Chain formula used was run in a MATLAB and the output was displayed graphically by using the functions in the MATLAB software. In this we have looked into similar studies done previously we have used 64 QAM which provides better performance compared to study conducted as explained in 7.2.1, the NS-3 C++ functions that we used were almost similar with most studies that we came across and are standard NS-3 function.

The delay of the WiMAX system in this studying has a similar trend or rather a relatively similar outcome, considering the type of programs used. In all programs the delay increased as the number of nodes increases, for NS-3 the delay increased as time (in seconds) increased. As much as the results are not exactly similar but all programs had a similar trend in delay. Retransmission probability and time also had an impact when it comes to the delay of the system. The throughput for the system also had a similar trend for all programs, the OPNET output increased as the number of nodes increase, time and retransmission probability had insignificant impact on the system when it comes to the delay for all programs used.

8.2 Future Work

WiMAX has opened a number of research possibilities as this is fairly new technology, implementation and the roll-out of the technology has dragged compared to LTE implementation. A study to integrate WiMAX to the existing LTE infrastructure is one of the areas that needs to be looked into. As the OPNET has surpassed other simulators for this research, it can then be used in the future study to simulate the integration of WiMAX with the existing LTE infrastructure. Both WiMAX and LTE are in the same family of broadband wireless network, integration seems logical and therefore can cut the cost of WiMAX implementation.
REFERENCES


41. B. Bose, I. Ahmad, and D. Habibii, "Bit error rate analysis in WiMAX communication at vehicular speeds using nakagami-m fading model." In Vehicular Technology Conference (VTC Fall), IEEE, pp. 1 – 5, 2012.


