

# **Discrete Event Simulation of Long Term Evolution Networks**

By

Jan Mikhail, B.Eng.

A thesis submitted to the Faculty of Graduate Studies and Research  
in partial fulfillment of the requirements for the degree of  
Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering (OCIECE)

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada, K1S 5B6

September 2016

© Copyright 2016, Jan Mikhail



# Abstract

Long Term Evolution-Advanced (LTE-A) is a novel mobile network standard that can address a number of challenges that network operators face while trying to support the growing demand for high data rates. To do so, LTE introduces a number of technologies including Coordinated MultiPoint (CoMP) and Device-to-Device (D2D) communication.

The performance of proposed protocols needs to be evaluated before the protocols are deployed on live networks. In fact, Modeling and Simulation (M&S) plays an important role in the development of modern cellular networks since it allows researchers to gain an insight into the operation of the networks in a cost and time effective manner. Discrete Event System Specification (DEVS) provides a formal platform for M&S of discrete event dynamic systems.

In this thesis, we present a general DEVS-based model for LTE networks. The model implements the basic functionality of each of the layers of the LTE protocol stack. In addition, it was designed to be flexible and modular. The model can be easily adapted to model various network deployments and scenarios, communication protocols, and propagation models.

Moreover, in this thesis, we present two novel algorithms that aim to improve the upload performance of UEs in LTE networks. The two algorithms, Shared Segmented Upload (SSU) and Upload User Collaboration (UUC), were developed in collaboration with fellow students at Carleton University, and Ericsson Canada. The algorithms rely on some of the technologies of LTE to enhance the upload process for UEs in the network, especially when the UEs are located at or near the edges of their cells.

The DEVS-based model we developed was used to conduct a series of system-level simulations to test the performance of the proposed algorithms, and compare them against the performance of traditional methods. The simulation results show that compared to the conventional methods, SSU improves the uplink performance for cell-edge UEs. In addition, UUC was shown to provide significant performance improvements for UEs regardless of their location within their cells, and can be applied to situations where CoMP is not available.

*Trust in the LORD with all your heart,  
And lean not on your own understanding;  
In all your ways acknowledge Him,  
And He shall direct your paths.*

*Proverbs 3:5-6*

# Acknowledgments

First, I would like to thank God for giving me the strength and patience to work on this thesis. It would not have been possible without His guidance and support.

I would like to thank my supervisor, Dr. Gabriel Wainer, whose invaluable advice and ongoing support have helped me tremendously over the duration of my graduate studies. The past few years have been a challenging and a rewarding experience.

The financial support of Carleton University and Natural Sciences of Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

I would also like to thank my colleague, Misagh Tavanpour, for his assistance in the collaborative work presented in this thesis. He provided phenomenal support during the past few years. Finally, I would also like to express my gratitude for Gary Boudreau of Ericsson Canada, who offered his expertise in the field of wireless communications and was always available to help.

# Table of Contents

|   |             |
|---|-------------|
| <b>Abstract</b>                                   | <b>iii</b>  |
| <b>Acknowledgments</b>                            | <b>v</b>    |
| <b>Table of Contents</b>                          | <b>vi</b>   |
| <b>List of Figures</b>                            | <b>viii</b> |
| <b>List of Tables</b>                             | <b>x</b>    |
| <b>List of Abbreviations</b>                      | <b>xi</b>   |
| <b>1 Introduction</b>                             | <b>1</b>    |
| 1.1 Overview                                      | 1           |
| 1.2 Contributions                                 | 4           |
| 1.3 Thesis Outline                                | 6           |
| <b>2 Background</b>                               | <b>8</b>    |
| 2.1 History of Mobile Communications              | 8           |
| 2.1.1 First Generation (1G)                       | 10          |
| 2.1.2 Second Generation (2G)                      | 11          |
| 2.1.3 Third Generation (3G)                       | 13          |
| 2.1.4 Fourth Generation (4G)                      | 20          |
| 2.2 Coordinated MultiPoint                        | 22          |
| 2.3 Modeling and Simulation for Cellular Networks | 26          |
| 2.4 Discrete Event System Specification           | 27          |
| 2.5 Related Work                                  | 30          |
| <b>3 Shared Segmented Upload</b>                  | <b>33</b>   |
| <b>3.1 Algorithm Overview</b>                     | <b>33</b>   |
| 3.2 SSU Messaging Structure                       | 38          |

|          |  |            |
|----------|--|------------|
| 3.2.1    | UploadRequest message                          | 39         |
| 3.2.2    | Handshake message                              | 39         |
| 3.2.3    | MetaInfo file message                          | 40         |
| 3.2.4    | Bitfield message                               | 40         |
| 3.2.5    | Piece message                                  | 41         |
| 3.2.6    | Done message                                   | 41         |
| 3.3      | Modeling of SSU using DEVS                     | 42         |
| 3.4      | Evaluation of SSU                              | 53         |
| 3.4.1    | Non-cooperative Algorithm                      | 53         |
| 3.4.2    | Propagation model and simulation parameters    | 53         |
| 3.4.3    | Simulation scenarios and results               | 55         |
| 3.5      | Summary  | 67         |
| <b>4</b> | <b>Upload User Collaboration</b>               | <b>68</b>  |
| 4.1      | Algorithm Overview                             | 69         |
| 4.2      | Modeling of UUC using DEVS                     | 76         |
| 4.3      | Evaluation of UUC                              | 87         |
| 4.3.1    | Non-cooperative Algorithm                      | 87         |
| 4.3.2    | Standard Coordinated MultiPoint Algorithm      | 87         |
| 4.3.3    | Propagation models and transmission parameters | 88         |
| 4.3.4    | Simulation scenarios and results               | 91         |
| 4.4      | Model Verification and Validation              | 100        |
| 4.5      | Summary  | 101        |
| <b>5</b> | <b>Conclusion and Future Work</b>              | <b>102</b> |
|          | <b>References</b>                              | <b>105</b> |

# List of Figures

|  |    |
|--|----|
| Figure 1.1: Effect of Inter-Cell Interference on cell-edge UEs. ....                     | 3  |
| Figure 2.1: Illustration of frequency reuse.....   | 10 |
| Figure 2.2: Evolution of the network architecture from UMTS to LTE [35]. ....            | 14 |
| Figure 2.3: Evolved Packet Core architecture [35].....                                   | 15 |
| Figure 2.4: E-UTRAN architecture [35].....   | 16 |
| Figure 2.5: E-UTRAN protocol stack [35]. ....  | 17 |
| Figure 2.6: LTE Resource Blocks [35]. ....   | 18 |
| Figure 2.7: Operation of a wireless relay in a HetNet [35].....                          | 21 |
| Figure 2.8: Coordinated Scheduling CoMP.....   | 24 |
| Figure 2.9: Coordinated Beamforming CoMP.....  | 24 |
| Figure 2.10: Joint Transmission CoMP. ....   | 25 |
| Figure 2.11: DEVS model diagram of a simple network.....                                 | 28 |
| Figure 2.12: Model file for a CD++ model of a simple network.....                        | 29 |
| Figure 3.1: A UE (UE1) wishes to upload a data file to the network.....                  | 35 |
| Figure 3.2: UE1 divides the data file into a number of pieces.....                       | 36 |
| Figure 3.3: The steps of a Shared Segmented Upload process. ....                         | 38 |
| Figure 3.4: DEVS model diagram for the mobile network model.....                         | 43 |
| Figure 3.5: Simplified segment of the Model file that defines the SSU DEVS model.....    | 45 |
| Figure 3.6: Simplified segment of the Model file that defines the SSU DEVS model.....    | 46 |
| Figure 3.7: Detailed UML class diagrams. ....  | 48 |
| Figure 3.8: UML class diagram for the messaging class hierarchy. ....                    | 49 |
| Figure 3.9: <i>UEProcessor</i> DEVS graph.....   | 51 |
| Figure 3.10: <i>BSProcessor</i> DEVS graph.....  | 52 |
| Figure 3.11: <i>UELink</i> state diagram. ....   | 52 |
| Figure 3.12: Simulation area showing eNB positions. ....                                 | 56 |
| Figure 3.13: Simulation scenario to evaluate SSU as a function of distance. ....         | 57 |
| Figure 3.14: Number of connected eNBs vs. distance from eNB (urban area, 900 MHz). ....  | 58 |
| Figure 3.15: Number of connected eNBs vs. distance from eNB (urban area, 2000 MHz). .... | 59 |
| Figure 3.16: File upload time vs. distance (urban areas, 900 MHz). ....                  | 60 |
| Figure 3.17: File upload time vs. distance (urban areas, 2000 MHz). ....                 | 60 |
| Figure 3.18: Data rate vs. Distance from eNB (urban area, 900 MHz). ....                 | 61 |
| Figure 3.19: Data rate vs. Distance from eNB (urban area, 2000 MHz). ....                | 61 |

|  |     |
|--|-----|
| Figure 3.20: Number of connected eNBs vs. distance from eNB (rural area).....                    | 62  |
| Figure 3.21: Average file upload time vs. distance from eNB (rural area). .....                  | 63  |
| Figure 3.22: Average data rate vs. distance from eNB (rural area). .....                         | 64  |
| Figure 3.23: Upload time vs. number of UEs (urban area, 900 MHz). .....                          | 65  |
| Figure 3.24: Upload time vs. number of UEs (urban area, 2000 MHz). .....                         | 66  |
| Figure 3.25: Data rate vs. number of UEs (urban area, 900 MHz).....                              | 66  |
| Figure 3.26: Data rate vs. number of UEs (urban area, 2000 MHz).....                             | 67  |
| Figure 4.1: UE1 wishes to upload a large data file. ....   | 69  |
| Figure 4.2: UE1 splits the data file into a series of pieces.....                                | 70  |
| Figure 4.3: UE1 assigns a set of pieces to each of the helper UEs.....                           | 71  |
| Figure 4.4: UE1 is no longer required to upload the entire file. ....                            | 72  |
| Figure 4.5: Initial steps of UUC (by the owner UE).....  | 74  |
| Figure 4.6: Upload of pieces by the owner UE.....  | 74  |
| Figure 4.7: Role of the helper UEs in the upload process. ....                                   | 74  |
| Figure 4.8: Process of gathering TBs and pieces.....   | 75  |
| Figure 4.9: Top-level model structure. ....  | 77  |
| Figure 4.10: Structure of the <i>UE</i> coupled model showing atomic and coupled submodels. .... | 79  |
| Figure 4.11: Structure of the <i>BS</i> coupled model showing atomic and coupled submodels. .... | 80  |
| Figure 4.12: Simplified UML class diagram for the model's atomic classes.....                    | 81  |
| Figure 4.13: DEVS graph for <i>UEPDCPLayer</i> . ....  | 82  |
| Figure 4.14: Format of the <i>sessionData</i> data structure. ....                               | 83  |
| Figure 4.15: UML class diagram showing the session classes. ....                                 | 84  |
| Figure 4.16: UML class diagram showing the algorithm class hierarchy. ....                       | 85  |
| Figure 4.17: <i>UETransmitterProcessor</i> DEVS graph.....                                       | 86  |
| Figure 4.18: <i>UEFilterProcessor</i> DEVS graph.....  | 87  |
| Figure 4.19: Network topology showing eNB positions. ....  | 92  |
| Figure 4.20: Average number of BSs per file vs. average distance from cell center.....           | 95  |
| Figure 4.21: Average upload time vs. average distance from serving eNB. ....                     | 96  |
| Figure 4.22: Average data rate of file upload vs. average distance from serving eNB. ....        | 97  |
| Figure 4.23: Average upload time per file vs. file size.....                                     | 98  |
| Figure 4.24: Average data rate per file vs. file size. ....                                      | 100 |

# List of Tables

|  |    |
|--|----|
| Table 3.1: <i>UploadRequest</i> message structure. ....                      | 39 |
| Table 3.2: <i>Handshake</i> message structure. ....                          | 39 |
| Table 3.3: <i>MetaInfo</i> file message structure. ....                      | 40 |
| Table 3.4: <i>Bitfield</i> message structure. ....                           | 41 |
| Table 3.5: Structure of a <i>Piece</i> message. ....                         | 41 |
| Table 3.6: <i>Done</i> message structure. ....                               | 42 |
| Table 3.7: Common simulation/transmission parameters. ....                   | 55 |
| Table 3.8: Simulation/transmission parameters specific for urban areas. .... | 55 |
| Table 3.9: Simulation/transmission parameters specific for rural areas. .... | 55 |
| Table 4.1: Common transmission parameters. ....                              | 90 |
| Table 4.2: Transmission parameters for uplink/downlink transmissions. ....   | 91 |
| Table 4.3: Transmission parameters for D2D transmissions. ....               | 91 |
| Table 4.4: Parameters for simulation scenario 1. ....                        | 93 |
| Table 4.5: Parameters for simulation scenario 2. ....                        | 98 |

# List of Abbreviations

|         |                                      |
|---------|--------------------------------------|
| 0G      | Zero Generation                      |
| 1G      | First Generation                     |
| 2G      | Second Generation                    |
| 3G      | Third Generation                     |
| 3GPP    | Third Generation Partnership Project |
| 4G      | Fourth Generation                    |
| AMPS    | Advanced Mobile Phone System         |
| BS      | Base Station                         |
| CB      | Coordinated Beamforming              |
| CDMA    | Code Division Multiple Access        |
| CoMP    | Coordinated MultiPoint               |
| CS      | Coordinated Scheduling               |
| CSI     | Channel State Information            |
| CSO     | Cell Switch Off                      |
| CSVD    | Cached and Segmented Video Download  |
| D2D     | Device to Device                     |
| DEVS    | Discrete EVent System Specification  |
| E-UTRAN | Evolved UTRAN                        |
| EIC     | External input couplings             |
| eICIC   | Enhanced ICIC                        |
| eNB     | Enhanced NodeB                       |
| EOC     | External output couplings            |
| EPC     | Evolved Packet Core                  |
| EPS     | Evolved Packet System                |
| FDMA    | Frequency Division Multiple Access   |
| GB      | Gigabytes                            |
| GPRS    | General Packet Radio Service         |
| HARQ    | Hybrid Automatic Repeat Request      |
| HetNets | Heterogeneous Networks               |
| HSS     | Home Subscriber Server               |
| HSPA    | High Speed Packet Access             |
| IC      | Internal couplings                   |
| ICI     | Inter-Cell Interference              |

|              |  |
|--------------|--|
| ICIC         | Inter-Cell Interference Coordination             |
| IMT-2000     | International Mobile Telecommunications-2000     |
| IMT-Advanced | International Mobile Telecommunications-Advanced |
| IMTS         | Improved Mobile Telephone System                 |
| IP           | Internet Protocol                                |
| ISI          | Inter-Symbol Interference                        |
| ITU          | International Telecommunications Union           |
| JT           | Joint Transmission                               |
| LAC          | Location-Aware Cooperation                       |
| LTE          | Long Term Evolution                              |
| LTE-A        | Long Term Evolution-Advanced                     |
| M&S          | Modeling and Simulation                          |
| MAC          | Medium Access Control                            |
| MCL          | Minimum Coupling Loss                            |
| MIMO         | Multiple Input Multiple Output                   |
| MME          | Mobility Management Entity                       |
| NAS          | Non Access Stratum                               |
| NMT          | Nordic Mobile Telephony                          |
| OFDM         | Orthogonal Frequency Division Multiplexing       |
| PDCP         | Packet Data Convergence Protocol                 |
| PDN-GW       | Packet Data Network Gateway                      |
| PHY          | Physical   |
| RAN          | Radio Access Network                             |
| RB           | Resource Blocks                                  |
| RLC          | Radio Link Control                               |
| RRC          | Radio Resource Control                           |
| S-GW         | Serving Gateway                                  |
| SAE          | System Architecture Evolution                    |
| SINR         | Signal-to-Interference-plus-Noise Ratio          |
| SMG          | Spatial Multiplexing Gain                        |
| SMS          | Short Message Service                            |
| SNR          | Signal-to-Noise Ratio                            |
| SSU          | Shared Segmented Upload                          |
| TACS         | Total Access Communication System                |
| TDMA         | Time Division Multiple Access                    |
| UE           | User Equipment                                   |
| UML          | Unified Modeling Language                        |
| UMTS         | Universal Mobile Telecommunications System       |
| UTRAN        | UMTS Terrestrial Radio Access Network            |
| UUC          | Upload User Collaboration                        |

# Chapter 1

# Introduction

## 1.1 Overview

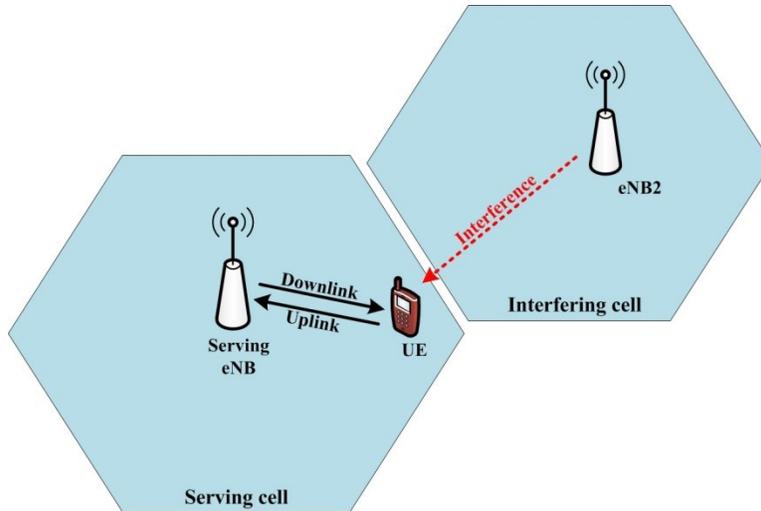
In the last few decades, mobile communication has evolved from being an expensive technology used on a small scale to the current more complex systems used on a large commercial scale. The use of mobile devices for communication has become part of our daily routine and has grown exponentially since the first mobile network was launched in 1983. By the end of 2015, the global number of mobile subscribers had surpassed 7.4 billion, a ten-fold increase from 738 million subscribers in 2000. This figure is expected to reach 9.2 billion by 2020 [1].

Mobile data was introduced in 2G communication systems, and since then, the availability of Internet access on mobile handhelds proved popular. During the period between 2010 and 2015, the global monthly data traffic generated by mobile phones has increased from 200 Petabytes to 3600 Petabytes [1]. This increase is not only driven by the growth in mobile subscriptions, but also an increase in the average data consumption per subscriber. Recent improvements in the performance of mobile networks as well as evolving smartphone capabilities and emerging data-intensive services have influenced the volume of data consumed by the average subscriber [2]. For example, Ericsson estimates that the average smartphone user consumed 1.4 Gigabytes (GBs) of data per month in 2015, up from 1 GB in 2014, and is forecasted to surpass 8.9 GBs per month by the end of 2021 [3].

Mobile service providers have been working considerably hard to create innovative solutions to meet the increasing consumer demand for higher data rates. Along with the increasing demand, mobile network operators face another challenge: the usable frequency spectrum, which all wireless communications rely on, is limited. Therefore, in order to meet the consumers' demands, service providers have to introduce new hardware and standards that make use of the available resources in a more efficient manner. To deal with these issues in the Fourth Generation (4G) of mobile systems, the Third Generation Partnership Project (3GPP) introduced the Long Term Evolution (LTE) and more recently, Long Term Evolution-Advanced (LTE-Advanced or LTE-A) standards. LTE-Advanced is an extension to LTE and was designed to be backwards compatible with LTE networks and devices. LTE-A meets or exceeds the requirements set by the International Telecommunications Union (ITU) in International Mobile Telecommunication-Advanced (IMT-Advanced), and is considered as a candidate for IMT-Advanced systems [4, 5, 6].

To maximize the frequency spectrum efficiency, LTE-Advanced has a unity frequency reuse factor, which means that neighboring base station nodes (called Evolved Node B (eNB) in LTE-Advanced) use the same frequency channels. This leads to severe interference problems between the signals of neighboring eNBs, known as Inter-Cell Interference (ICI). The effect of ICI on the data rate is magnified at the cell edges where the distance between the User Equipment (UE) and the interfering eNB is small, as seen in figure 1.1. Moreover, cell-edge users receive lower signal strength from their cell's serving eNB due to the larger distance between them. Both of these factors cause significant degradation of performance for users along the cell borders [7].

Enhanced Inter-Cell Interference Coordination (eICIC) and Coordinated MultiPoint (CoMP) are two important technologies employed by LTE-Advanced to cope with the interference issues, especially for cell-edge users. In general, CoMP refers to a range of techniques that make use of a set of eNBs, called the coordination set, that dynamically coordinate the transmission and reception of data to minimize the effects of interference. Essentially, CoMP turns ICI into a useful signal that could be used to improve the performance of cell-edge UEs [7].



**Figure 1.1: Effect of Inter-Cell Interference on cell-edge UEs.**

To meet the continually increasing demand for high data rate services, large high-speed networks have been designed and used. Modern network systems employ a fairly large number of complex standards. As new protocols are proposed, there is a strong need to evaluate the performance and efficiency of these protocols under various conditions before they are standardized. Testing complex protocols on live networks is not suitable. The use of an adequate modeling and simulation (M&S) technique allows network designers to have better insights into the operation of the network while avoiding the costs of modifying existing networks [8]. A suitable M&S platform provides researchers with an environment to model a mobile network with a desirable level of accuracy, and to run simulations that test the performance of the proposed protocols. This process reduces the cost and time required to develop new standards. It also reduces the risk accompanied with testing protocols on live networks.

There are a number of M&S tools available for use in the field of communication networks. NS-2, NS-3, OPNET, OMNeT++, and JiST are some of the popular simulators used in this field [9, 10]. These simulators provide strong platforms for researchers to model a system and test proposed techniques under various conditions. However, each of these simulators has drawbacks as well. The strengths and weaknesses of these simulators are discussed in Chapter 2. In order to avoid these drawbacks, and to introduce a flexible and efficient M&S technique that can be used in the field of communication networks, we have used Discrete Event System Specification [11].

Discrete Event System Specification (DEVS) is a general, modular, and hierarchical formal framework for modeling and simulation. Using DEVS, a complex system can be defined using a set of *atomic* and *coupled* components [11, 12]. Atomic models are the “building blocks” of the system and define the behaviour of the system’s components while coupled models define the hierarchy of the system. Coupled models are composed of one or more atomic or coupled models, along with the interconnections between them. The DEVS formalism is presented in detail in Chapter 2.

## 1.2 Contributions

In this thesis, we present a new LTE model based on Discrete Event System Specifications (DEVS). In the model, we implemented the basic functions of the LTE protocol stack in both UE terminals as well as eNBs. Moreover, the model supports various radio propagation mechanisms in different cell deployment scenarios, and it was designed in such a way that it would be easy to adapt for simulating other algorithms and techniques. The model is presented in detail in Chapter 4.

In addition, we present two advanced algorithms, namely Shared Segmented Upload (SSU) and Upload User Collaboration (UUC). Both algorithms are based on a distributed CoMP architecture, and aim to improve the uplink data rate for users uploading large files to a mobile network. The first algorithm, SSU, was developed in collaboration with Mohammed Moallemi (former Postdoctoral fellow at Carleton University), and Misagh Tavanpour (PhD candidate at Carleton University). The algorithm was initially proposed by M. Moallemi and later refined by M. Tavanpour and I. UUC was developed in collaboration with M. Tavanpour who proposed the idea. In both algorithms, I assisted in the design and refinement of the messaging protocols, planning of the simulation scenarios, as well as the collection and analysis of the results. Both algorithms have been developed in collaboration with Gary Boudreau and Ronald Casselman from Ericsson Canada, who helped with the validation process during the simulation activities. Both algorithms have been recently patented [13, 14].

SSU works by dividing the file into a number of pieces of fixed length. The pieces are then uploaded to the UE's coordination set of eNBs, simultaneously. SSU is based on the same fundamental concepts as the BitTorrent protocol [15]. BitTorrent is a peer-to-peer protocol used for distributing large files between users on the Internet and it works by allowing users to join "groups" of hosts that can exchange parts of the files concurrently. This technique was adapted and used by SSU in such a way that users close to a cell's edge can upload pieces of the file to multiple eNBs independently on different frequency channels in parallel, thus improving the overall upload rate. The pieces are then collected by the UE's serving eNB and the original file is reconstructed.

UUC shares a few similarities with SSU. UEs employing UUC upload files in a similar manner by dividing the file into pieces with varying sizes and uploading the pieces to multiple eNBs simultaneously. However, UUC also makes use of LTE's Device-to-Device (D2D) capabilities by allowing UEs to request assistance with the file upload from neighboring UEs. Parallel to the UE-to-eNB upload, the UE is able to share some of the file pieces with other UEs, who in turn, upload these pieces on their own uplink channels to the network. This process promises significant improvements to the uplink data rate of UEs regardless of their position within the cell. Moreover, UUC exploits the joint reception of pieces by the eNBs in a coordination set to improve the efficiency of LTE's retransmission mechanism.

Additionally, a conventional non-cooperative algorithm and a standard CoMP algorithm were implemented in order to evaluate the performance of each of the proposed algorithms in a number of scenarios. The scenarios were simulated under two settings: rural and urban areas. The simulations show that both algorithms provide a significant improvement in the uplink data rates, especially for users at or near the cell edge.

There are various publications derived from this research:

- [16] M. Tavanpour, J. Mikhail, M. Moallemi, G. Wainer, G. Boudreau, and R. Casselman, "Data upload in LTE-A mobile networks by using shared segmented upload," *J. Networks*, vol. 10, no. 4, pp. 252–264, 2015.

- [17] M. Tavanpour, J. Mikhail, G. Wainer, M. Moallemi, G. Boudreau, and R. Casselman, “DEVS Based Modeling of Shared Segmented Upload in LTE-A Mobile Networks,” in Proceedings of the 18th Symposium on Communications & Networking, 2015, pp. 60–67. Alexandria, VA, USA.
- [18] M. Tavanpour, M. Moallemi, G. Wainer, J. Mikhail, G. Boudreau, and R. Casselman, “Shared segmented upload in mobile networks using coordinated multipoint,” in Proceedings of the 2014 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2014 - Part of SummerSim 2014 Multiconference, 2014, pp. 662–669. Monterey, CA, USA

In addition, the collaboration with Ericsson has resulted in the following two patents.

- [13] M. Tavanpour, J. Mikhail, G. A. Wainer, G. Boudreau, H. Seyedmehdi, and R. Casselman, “File Transfer by Mobile User Collaboration”, Provisional Patent Filing Reference P46444 WO1. June 2015. PCT/IB2015/054524. USA.
- [14] M. Tavanpour, M. Moallemi, J. Mikhail, G. A. Wainer, G. Boudreau, and R. Casselman, “Shared Segmented Upload in Mobile Networks using Coordinated Multipoint”, Provisional Patent Filing Reference P43130 US1. February 2015. PCT/IB2015/051404. USA.

### 1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides a brief history of mobile communications and lays the foundation background theory on the wireless technologies used as a basic for the work presented in this Thesis. A literature review related to advancements in different aspects of Coordinated Multipoint with the aim of increasing data rates for cell-edge users is also presented in Chapter 2. Chapters 3 and 4 present the two proposed algorithms, Shared Segmented Upload (SSU) and User Upload Collaboration (UUC), respectively. Both chapters discuss the algorithms in detail and present the complete DEVS models used to simulate them. The results of the simulation and an analysis of the performance of the algorithms are also

discussed in these chapters. Finally, Chapter 5 summarizes the contributions and results obtained in this Thesis. Additionally, this chapter proposes a number of possible future extensions to the work presented.

## Chapter 2

# Background

This chapter introduces some of the key mobile communication systems of the past and presents the fundamental technologies that power today's modern systems. Section 2.1 offers a brief history of mobile communication technologies, and an overview of Long Term Evolution-Advanced and the techniques it employs to overcome the problems faced by older systems. Moreover, in section 2.2, we describe Coordinated Multipoint, the underlying technology on which the presented work is based, in detail. The following section highlights the importance of modeling and simulation in the field of communication while section 2.4 discusses other research efforts related to CoMP and the improvement of quality of service for cell-edge users. Finally, in section 2.5, we define the DEVS formalism used to model and simulate the proposed algorithms.

### 2.1 History of Mobile Communications

James Clerk Maxwell laid the foundations of radio communications in 1864 with his theory of electromagnetic radiation. Maxwell's equations, as they are known today, defined how electromagnetic waves propagate in free space [19]. Later on, in 1897, Guglielmo Marconi experimented with radio communication and was soon successful in transmitting a wireless telegraph message over a distance of 2.8 kilometers [19, 20]. Experiments and trials with radio communication continued into the early 20<sup>th</sup> century and resulted in the launch of the first commercial telegraphy service in 1907.

Modern mobile communications have evolved greatly since the first telegraphy systems, both in complexity and performance. Advancements in electrical circuitry and hardware design have led to devices that are more portable than previous telegraphy systems, and are capable of two-way voice transmissions. Soon after, the first commercial mobile telephone systems were launched in the United States in the late 1940s, and in Europe in the 1950s. These systems were simple but proved the popular demand for wireless communications, and laid the foundations for future systems. These systems are usually referred to as 0G (zero generation) systems [21].

The first mobile telephone systems before the 1990s were analog and were based on Frequency Division Multiple Access technology (FDMA). The mobile terminals were bulky and heavy, and therefore, not very portable. Moreover, they relied on base stations that were able to transmit powerful signals that provide coverage over large areas, typically 30 to 50 kilometers [22]. However, to avoid interference, the reuse of any frequency channel for a different connection required large separation distances. For this reason, along with the fact that the available frequency bandwidth is limited, the capacity of these systems was small. Most of the available frequency spectrum was dedicated to emergency and public services, while the remaining spectrum was available to a very limited number of users. For example, in 1965, the Improved Mobile Telephone System (IMTS) operating in New York City had 2,000 customers, who shared 12 channels, and often had to wait 30 minutes to place a call [23].

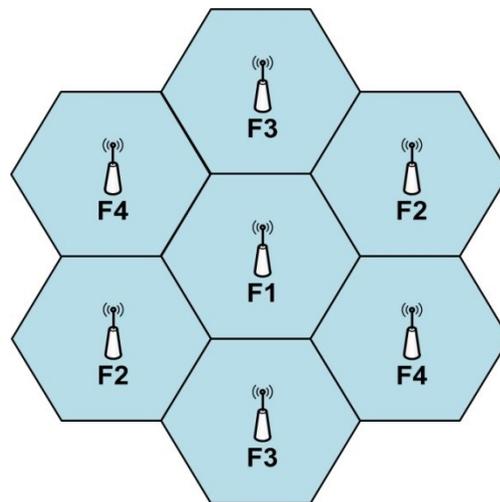
As the demand for mobile services increased and penetrated new markets, service providers in different regions launched their own systems and these systems were incompatible with each other. The development of telephony systems was fragmented, and mostly owned and operated by companies that had close relationships with their governments [24]. To offer services on a larger scale, the mobile communications industry realized the need to develop standards. These standards were developed with the goal of merging existing technologies and offering fewer services that are more robust. These standards are often grouped into generations and referred to as 1G for First Generation, 2G for Second Generation, and so on [7].

### 2.1.1 First Generation (1G)

The first generation of mobile communication systems was also based on analog transmission. However, these systems were designed solely for voice services, and relied on circuit-switched technology.

As discussed in the previous section, 0G systems had limited capacity; they relied on one powerful Base Station (BS) that covered a large geographical area with limited coverage depending on their transmission power. Moreover, due to the limited availability of usable spectrum, these systems were only able to serve a small number of users within the coverage area. To increase the capacity of mobile communication systems, V. H. Mac Donald proposed in 1979 a cellular system, which has hexagonal cells covering an area with slight overlaps, in a paper titled “The Cellular Concept” [25]. The concept was a major step in the advancement of mobile communication systems and overcame many of the challenges in wireless systems, such as power consumption, coverage, capacity, spectral efficiency, and interference.

The key concept of cellular networks is *frequency reuse*, allowing different cells in the network to be assigned different frequency bands in such a way that adjacent cells do not use the same frequencies [25]. Figure 2.1 demonstrates the concept of frequency reuse.



**Figure 2.1: Illustration of frequency reuse.**

The different sets of frequencies used in each cell are labelled  $f_1$  to  $f_4$ . In this setup, a communication session between a Base Station (BS) and a User Equipment (UE) would be “handed off” to a neighboring BS when the user moves across the cell boundary. This approach reduces the interference from neighboring cells, improves the spectral efficiency, and more importantly, increases the capacity of the system allowing it to serve a larger number of customers simultaneously. Moreover, since the cells are smaller, the BS antennas transmit at lower power and consume less power [25, 26]. This also leads to phones that are more portable with smaller and lighter batteries.

As the demand for frequency channels increases within a given cell beyond its capacity, the overloaded cell is “split” into a number of smaller cells, each with its own BS. The frequencies used in the original cell are then reallocated to the smaller cells. This process is called cell splitting and gives network operators the flexibility to meet the increasing demand in urban areas [27].

The first 1G cellular systems were designed in the late 1960s. However, due to regulatory delays, they were launched in the 1980s. The first trials of a fully operational system were conducted in Chicago in 1978. Advanced Mobile Phone System (AMPS) was the first analog commercial system to be launched in the United States in 1982. Soon after, similar systems were launched in different parts of the world, such as Total Access Communication System (TACS) in the United Kingdom, Italy, Spain, Austria and Ireland, and Nordic Mobile Telephony (NMT) in the Nordic countries and Russia [28].

### 2.1.2 Second Generation (2G)

Analog 1G systems were the first step in the mobile communication industry. Despite its success, analog systems had a number of disadvantages that limited their performance and capacity. The second generation of mobile telephony systems alleviated many of these problems by transmitting data with digital signals. Compared to analog systems, digital systems have a number of advantages.

Firstly, analog signals are difficult to encrypt. It was easy to intercept and listen to a conversation using simple techniques and relatively cheap hardware. In fact, this was a common security issue with 1G cellular systems since user identification numbers were often intercepted and used to make calls illegally. On the other hand, digital signals are easy to encrypt in order to improve security and privacy [28].

Secondly, error detection and correction schemes can be applied to digital signals. These techniques result in more robust and reliable transmissions since the receiver can detect and possibly correct erroneous bits. In contrast, interference or noise in an analog system results in inconsistent voice quality [28, 29].

Lastly, in analog systems, a pair of radio channels are dedicated to each duplex call, even when users are inactive (not speaking). In digital systems, radio channels are shared between users and are assigned using different time slots (Time Division Multiple Access, or TDMA, systems) or by using different coding schemes (Code Division Multiple Access, or CDMA, systems). Users are assigned time slots or codes only when they have voice or data to transmit. This increases the spectral efficiency and the overall capacity of the system [28].

In addition to the benefits of digital transmissions, 2G introduced a number of services including caller identification, call forwarding and Short Message Service (SMS). SMS uses a set of standardized protocols to allow users to exchange short text messages, up to 160 characters. Despite this limitation, SMS became extremely popular. By the end of 2010, SMS was being used by an estimated 3.5 billion subscribers, or 80% of all mobile phone subscribers [30].

2G systems relied on a circuit-switched domain which dedicates a complete communication channel to a given user. This approach is usually inefficient since the communication channel often only carries data for a small percentage of the time. In contrast, packet-switched systems are more efficient since a communication channel is only occupied by a user when data is being transmitted, allowing multiple users to share a single communication channel. General Packet Radio Service, or GPRS, is a mobile data service that relies on a packet-switched domain. GPRS was introduced to 2G systems to provide Internet communication

services at a moderate transfer speed of 56–114 Kbits/second. 2G mobile networks that employ GPRS are often informally referred to as 2.5G systems [7, 31, 32].

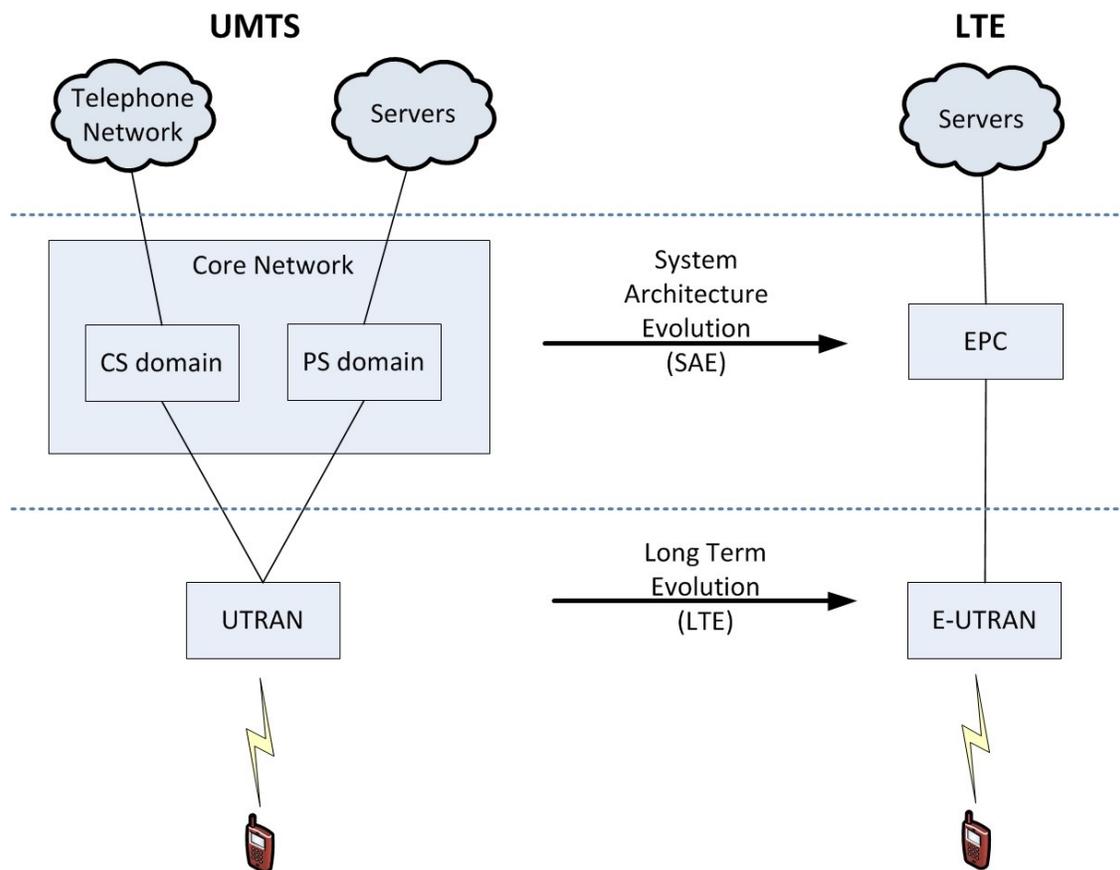
### 2.1.3 Third Generation (3G)

In the earlier days of mobile communications, several organizations were tasked with developing technical standards in their respective regions. However, the rapid growth of mobile networks in the late 1990s prompted the International Telecommunications Union (ITU) to publish a set of requirements for a 3G mobile communication system to drive the development of 3G technologies. The requirements were released under the name International Mobile Telecommunications-2000 (IMT-2000) [33]. In order to facilitate the development of a unified 3G standard, a collaboration project between seven standards organizations was initiated. The project, called 3rd Generation Partnership Project (3GPP), is now responsible for developing and maintaining telecommunication standards for global use [31].

The outcome of the standardization efforts by 3GPP for 3G systems were released as a set of technologies that met the requirements defined in IMT-2000. The main objectives of IMT-2000 systems were to provide global roaming, allowing users to move freely across borders while using the same number and handset, support a wide range of services including voice, data, Internet, and multimedia services, as well as provide higher data rates of up to 2 Mbits/second for stationary users [33]. Systems that meet these requirements are referred to as 3G systems. A number of 3G networks have since been introduced, including Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), and Long Term Evolution (LTE). HSPA and LTE are considered to be evolved standards based on the UMTS standard, and were made available in 3GPP's release 5 and release 8, respectively. Each of these releases has improved the overall performance of previous releases, while focusing on increasing peak data rates, reducing latency, and increasing the capacity of the system [7].

The development of LTE systems was mainly driven by the creation of new services for mobile devices and the need for a more capable network infrastructure that is able to deliver higher data rates at lower costs [34]. Figure 2.2 shows the evolution of the new system architecture from that of UMTS. LTE was focused on the development of a new radio access

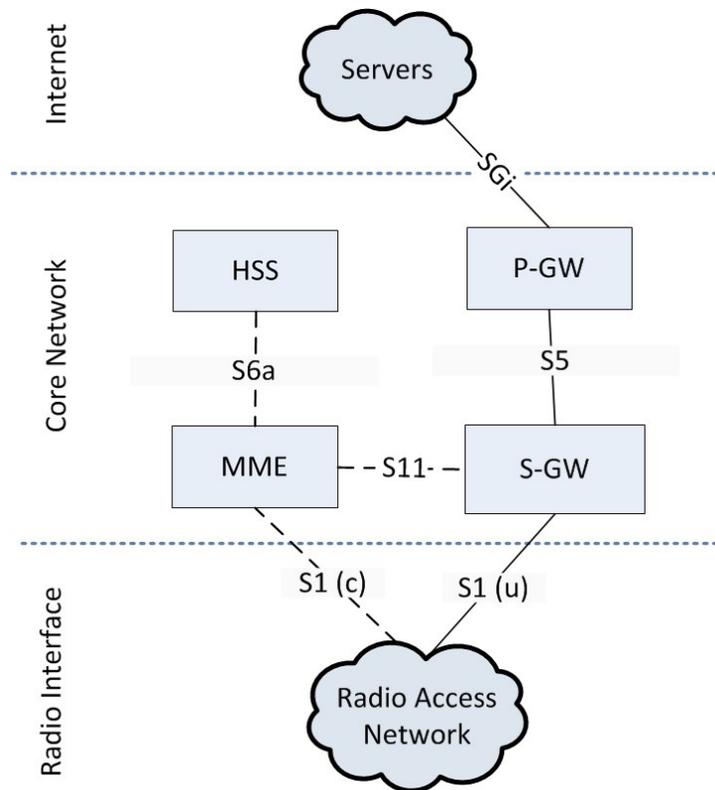
network and air interface with the assumption that all communication services, including voice services, would be based on a packet-switched model. The resulting radio interface is called Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), which replaces UMTS Terrestrial Radio Access Network (UTRAN) in earlier UMTS networks. LTE was accompanied by an evolution, referred to as System Architecture Evolution (SAE), of the non-radio core network into the Evolved Packet Core (EPC). Both LTE and SAE are part of the Evolved Packet System (EPS). The term LTE has since been used informally to refer to the whole system, and will be used in a similar manner for the remainder of this thesis [35].



**Figure 2.2: Evolution of the network architecture from UMTS to LTE [35].**

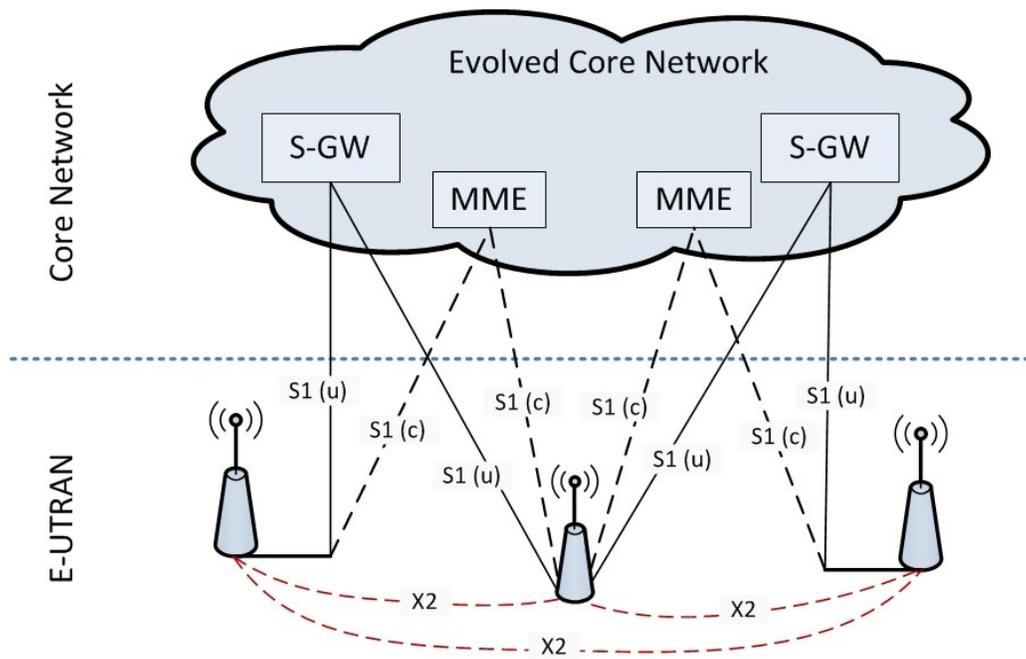
LTE’s Evolved Packet Core (EPC) is an all-IP-based packet-only switched network, unlike UMTS’ core network, which relies on circuit switching for voice services (and packet switching for other data services). EPC is responsible for all non-radio related functionality of the LTE system and is composed of four main components: Mobility Management Entity

(MME), Serving Gateway (S-GW), Packet Data Network Gateway (PDN-GW), and Home Subscriber Server (HSS) [36]. Figure 2.3 shows the high-level system architecture of LTE's EPC. The MME is a key element in the control plane of LTE networks. It is responsible for managing security functions such as authentication and authorization, mobility functions, roaming, and coordinating handovers, among other functions. The S-GW lies in the network's user plane and is the point at which the EPC connects to the radio access network (RAN). The S-GW is responsible for managing and transferring user data between the eNBs in the radio access network and the PDN-GW, which serves as a gateway to the internet. The PDN-GW serves as a gateway to the Internet and is responsible for the IP address allocation to UEs. When a UE device is switched on, the serving BS contacts the MME that, in turn, authenticates the user and requests an IP address from the PDN-GW. Finally, the HSS is a database that holds subscription-related, subscribers' locations and IP addresses, etc [35].



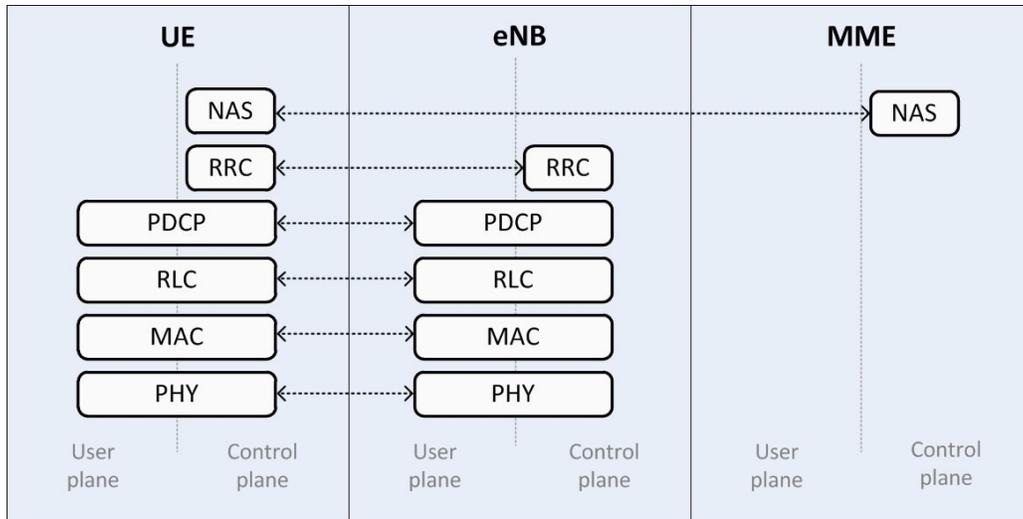
**Figure 2.3: Evolved Packet Core architecture [35].**

Evolved UMTS Terrestrial Radio Access, or E-UTRAN, is LTE’s radio access network and is responsible for all radio-related functionality of the network including scheduling, modulation, coding, transmission, etc. E-UTRAN is composed of one type of node, called evolved NodeB, or eNodeB (eNB), which provides the air interface between the network and the UEs. Each of the eNBs is a logical node that serves one more cells, and the interface interconnecting the eNBs is called the X2 interface [35, 37]. Figure 2.4 illustrates the architecture of an E-UTRAN network. The eNBs transfer packets to and from the nodes in the core network through the S1 interface.



**Figure 2.4: E-UTRAN architecture [35].**

The communication links between the eNBs and the nodes in the core network as well as the links between the eNBs and UEs rely on the E-UTRAN protocol stack. The protocols can be logically divided into control plane (responsible for managing the transport/link bearer) and user plane (responsible for transferring user data). In both the user and control planes, the protocols that are included are the Packet Data Convergence Protocol (PDCP), the Radio Link Control (RLC), Medium Access Control (MAC), and the Physical Layer (PHY) protocols. Additionally, the Non Access Stratum (NAS) and Radio Resource Control (RRC) protocols are only included in the control plane. The protocols defined in LTE’s E-UTRAN network are shown in Figure 2.5.

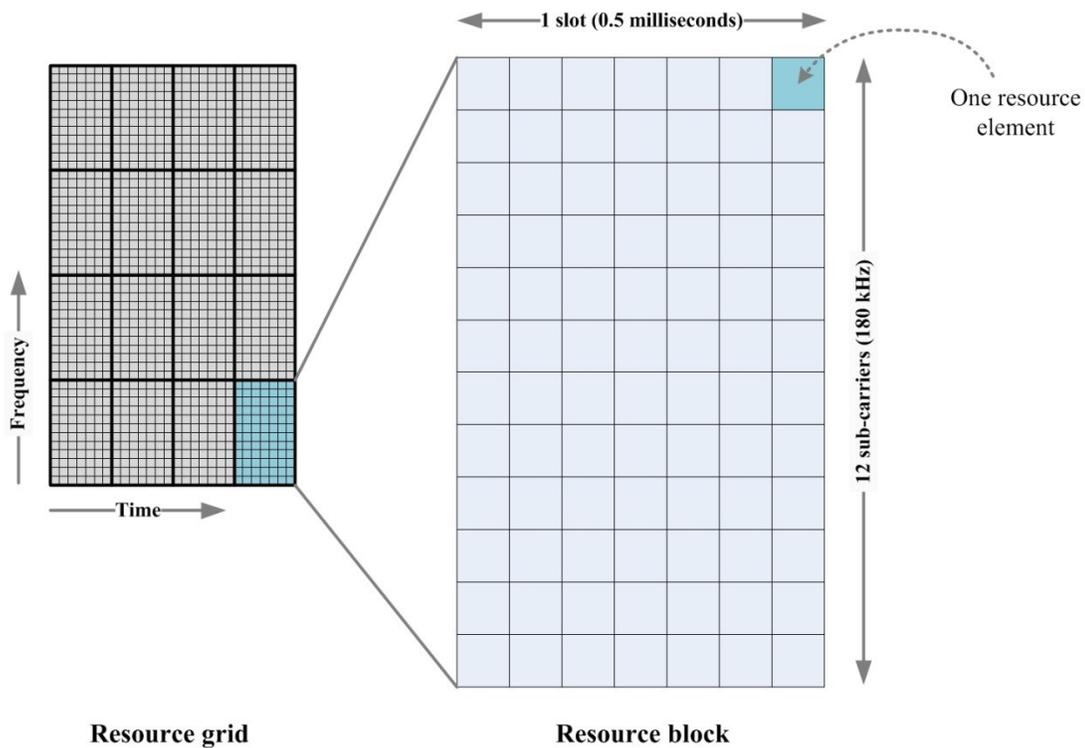


**Figure 2.5: E-UTRAN protocol stack [35].**

The NAS layer is responsible for managing connections and sessions between the UE and the core network (more specifically, the MME) as well as authentication and registration of subscribers. The main functions of the RRC layer are to establishment, management, and release of RRC connections, broadcast of system information and radio bearer establishment. The RRC layer provides a medium for the transmission of NAS messages between the UE and the MME, through the eNBs [35, 37]. The PDCP layer handles IP header compression, ciphering, in-sequence delivery, duplicate detection, etc. The RLC layer is responsible for segmentation/concatenation of data packets, reordering for in-sequence delivery, error detection and correction, as well as retransmission of erroneous packets. The MAC layer is responsible for link adaptation and the multiplexing/de-multiplexing of logical channels onto physical transport channels, error correction using Hybrid Automatic Repeat Request (HARQ), and scheduling [34, 37]. The PHY layer, as the name suggests, deals with all the details related to the actual transmission of packets to and from the node over the radio interface. It handles coding/decoding, modulation/demodulation, multi-antenna mapping, among other typical physical layer functions [35, 36, 38].

A Scheduler is a component of LTE eNBs that is responsible for dynamic scheduling, a function that deals with the question of how to share the limited radio resources among users in a fair and efficient manner while achieving the required quality of service. In LTE, the scheduling

operations are performed every millisecond and the scheduling decisions are then forwarded to the UEs. The transmissions are organized as a function of frequency as well as time, as illustrated in Figure 2.6. Transmissions can be scheduled by Resource Blocks (RB), each of which consists of 12 sub-carriers (with a total of 180 kHz) in the frequency domain, and a duration of 0.5 milliseconds (one slot, or 7 symbols) in the time domain. A Resource Block is further divided into smaller units called Resource Elements. A Resource Element occupies one symbol in the time domain and one subcarrier in the frequency domain. Thus, a Resource Block consists of 84 Resource Elements. The scheduler in an eNB uses Resource Blocks for frequency-dependent scheduling, by allocating Resource Blocks to UEs, in both the time and frequency domain [35].



**Figure 2.6: LTE Resource Blocks [35].**

The design of LTE was mainly driven by the demand for higher data rates while increasing the capacity of the network. LTE was required to deliver peak data rates of 100 Mbps in the downlink and 50 Mbps in the uplink, a 7-fold increase over the peak data rates of HSPA, while supporting a minimum of 200 active subscribers in each cell with 5 MHz bandwidth [34].

Low latency was another important design target, especially for voice and other delay sensitive applications. The requirements stated that, in ideal conditions, the time needed for data to travel from a UE to an eNB and back should be under five milliseconds, compared to HSPA's eighty-millisecond latency. LTE introduced a number of techniques in order to meet these goals. Three important techniques are briefly described below.

- Orthogonal Frequency Division Multiplexing (OFDM): OFDM is a frequency-division multiplexing scheme used as a multi-carrier modulation method, in which data is encoded onto multiple sub-carrier frequencies. Each sub-carrier is only required to deliver a portion of the overall data rate, which reduces the overlap between the symbols. Moreover, the sub-carriers are selected in such a way that they do not interfere with each other. This leads to a reduction in Inter-Symbol Interference (ISI) and reduces the probability of errors at the receiver [28].
- Inter-Cell Interference Coordination (ICIC): In order to increase the capacity of the system, LTE uses a frequency reuse factor of one. This means that each eNB in the network is assigned the same set of frequencies, which leads to a higher level of interference between neighbouring eNBs, especially for users at the cell edges. This form of interference is referred to as Inter-Cell Interference, or ICI. ICIC can reduce such interference by allowing neighbouring eNBs to exchange interference information through the X2 backhaul. The eNBs can then use the interference information to coordinate the scheduling decisions in order to limit the effects of ICI [35].
- Multiple antenna techniques: LTE allows the use of multiple antennas at the transmitter, the receiver, or both. The use of multiple antennas is beneficial in a number of ways. First, in a method called Receive Diversity, a receiver can use two or more antennas to detect multiple versions of the received signal. These signals reach the antennas with slightly different phase shifts. The receiver can add the received signals together to remove the destructive effects of multi-path fading. Transmit Diversity works in a similar manner, but uses multiple antennas at the transmitter side. The transmitter sends multiple versions of the same signal with specific differences in their phase shifts, such that they reach the receiver antenna in

phase. Multiple antennas can also be used to concentrate the signal from an eNB in a certain direction using a technique called Beamforming. This technique allows multiple users located in different directions to be served simultaneously. Finally, a transmitter with multiple antennas can transmit different streams of data in separate spatial dimensions using a technique called Spatial Multiplexing Gain (SMG). This results in an increase in the capacity of the system without requiring additional bandwidth. In LTE, this technique is referred to as Multiple Input Multiple Output, or MIMO [35, 39].

Since the launch of the first LTE network in 2009, LTE has seen rapid growth in the number of deployments. By the beginning of 2016, 480 LTE networks were operational in 157 countries, serving over 1.1 billion subscribers. The number of subscribers is expected to increase to 4.3 billion by the year 2021 [3, 40].

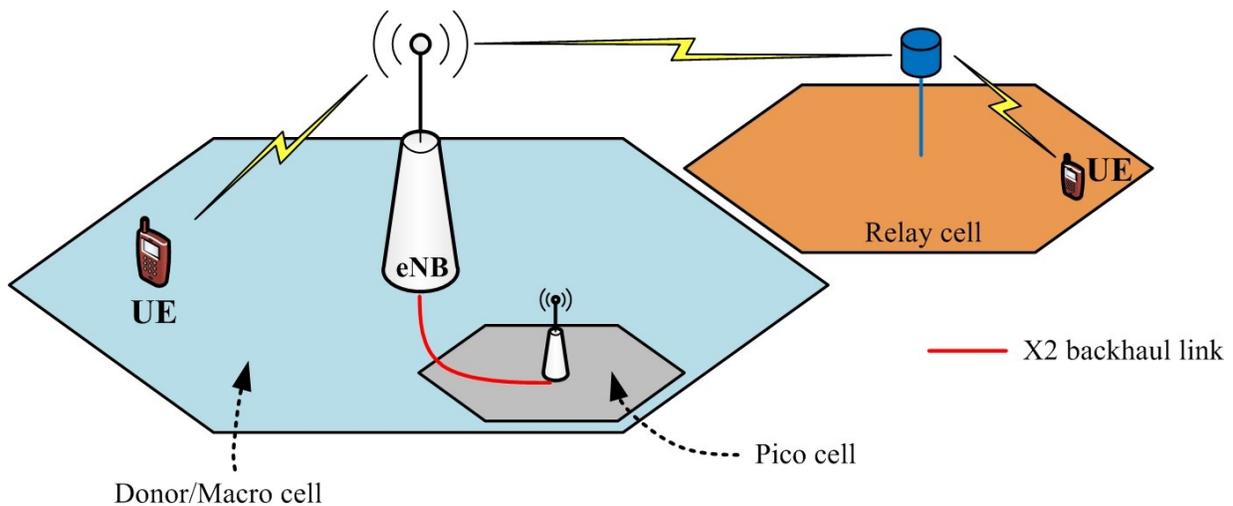
#### 2.1.4 Fourth Generation (4G)

As previously mentioned, the design of 3G technologies was driven by IMT-2000, which defined a set of requirements for 3G communication systems. In 2008, the ITU launched a similar process and published a set of requirements for 4G networks under the name IMT-Advanced [41]. These requirements state that the peak data rate for 4G systems should be at least 600 Mbps in the downlink and 270 Mbps in the uplink, using a maximum bandwidth of 40 MHz.

One of the most promising candidates for 4G networks is LTE-Advanced (LTE-A). LTE-A was designed to exceed the minimum requirements set by the ITU in IMT-Advanced, while maintaining backward compatibility with older eNBs and UEs designed for LTE networks. However, 3GPP had set more demanding performance targets for LTE-A. LTE-A was required to deliver peak data rates of 1000 Mbps in the downlink and 500 Mbps in the uplink [34, 39].

To reach these performance targets, LTE-A utilizes a set of technologies that include carrier aggregation, improved Multiple Input Multiple Output (MIMO) techniques, relays, heterogeneous deployments, enhanced ICIC (eICIC), and Coordinated MultiPoint (CoMP). These technologies are briefly described below.

The most straightforward way to increase the capacity of a network is to utilize more bandwidth. However, the available spectrum is often fragmented and a single wideband spectrum is rarely available to a single service provider. LTE-A introduced a method called carrier aggregation, in which multiple carriers of different bandwidths are aggregated and collectively used for one transmission. To maintain backward compatibility with LTE, each component carrier can have a bandwidth of up to 20 MHz, and up to 5 component carriers can be aggregated to a total of 100 MHz [39, 42]. LTE-A's MIMO expands on the special multiplexing technique used in LTE by allowing the use of 8 transmit and 8 receive antennas in the downlink, and 4 transmit and 4 receive antennas in the uplink [42]. Relays are employed by LTE-A to extend the coverage of a network into remote areas without the need for expensive X2 backhaul connections. A relay node is a low-power node that connects to a neighbouring eNB (called donor cell) using radio channels and appears as an ordinary cell to a UE. Similar to the concept of relays, LTE-A supports the deployments of eNBs that have different transmit power and geographical coverage within the same network. Such networks that include a mixture of high-power macro cells and low-power pico cells are called heterogeneous networks, or HetNets. This approach delivers high per-user capacity and high data rates in areas covered by the pico cells, as well as increase the performance of the macro cells by offloading traffic generated within the pico cells [39]. Figure 2.7 shows a HetNet network with one macro eNB, one pico eNB, and one relay node.



**Figure 2.7: Operation of a wireless relay in a HetNet [35].**

Enhanced ICIC is an advanced version of ICIC, first introduced in LTE networks, evolved to support the use of HetNets. In HetNets, interference is a more significant challenge where low-power pico nodes are deployed within a high-power macro cell's coverage area. Coordinated MultiPoint is another technique employed by LTE-A to combat the effects of interference between cells, especially for users at the cell edges. CoMP allows a number of eNBs to dynamically coordinate the communication with a UE in order to increase the signal strength received by the UE, or reduce the interference caused by neighbouring cells [43, 44]. Section 2.2 describes the operation of CoMP in more detail.

## 2.2 Coordinated MultiPoint

As previously mentioned, in order to increase the capacity of the system, LTE uses a frequency reuse factor of one, which means that the entire frequency spectrum is utilized by each eNB in the network. Hence, LTE macro cells experience high levels of interference, known as Inter-Cell Interference (ICI), between neighbouring eNBs that communicate with their UEs on the same frequency bands. This type of interference is more severe for UEs located close to the edges of their cells. Moreover, due to the distance-dependent path losses and the larger distances between these UEs and their serving eNBs, these UEs receive weaker signals from their serving eNBs. As a result, cell-edge users often experience the lowest data rates within the cell [7].

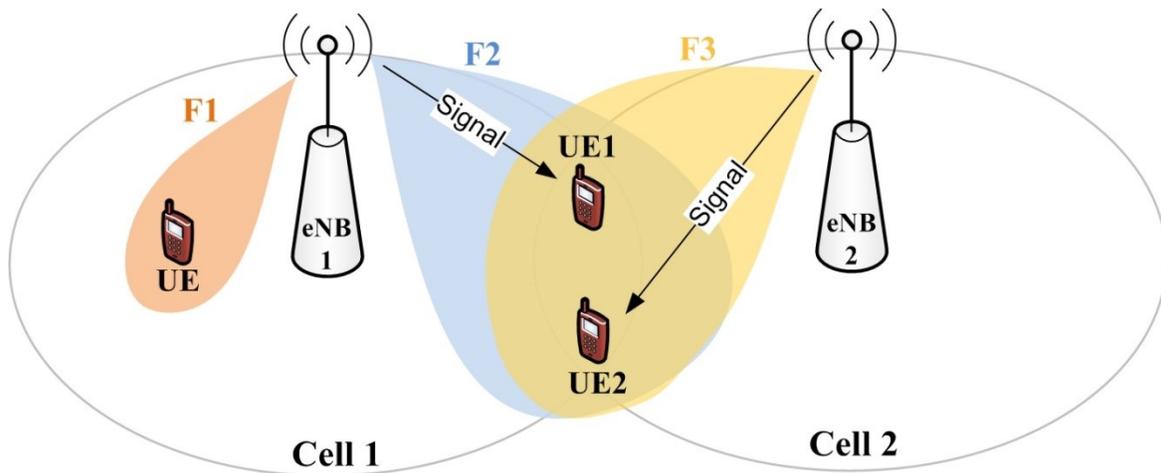
One of the key targets of LTE-A is to provide consistent performance for the UEs regardless of their location. To achieve this, LTE-A introduces a technique called Coordinated MultiPoint (CoMP) which allows a number of eNBs to coordinate the communication with a UE dynamically in order to reduce the level of interference, as well as increase the signal strength received by the UE. In essence, CoMP allows a UE at the edge of a cell to communicate, not only with its serving eNB, but also with other eNBs in a coordinated manner to increase the overall throughput available to the UE. A group of coordinating eNBs in CoMP are referred to as the coordination set. This type of coordination requires the exchange of certain control information between the eNBs. This control information includes scheduling decisions and Channel State Information (CSI). UEs measure the quality of their channels and relay the

resulting information to their serving eNBs, which in turn, exchange it with neighbouring eNBs in the coordination set [7].

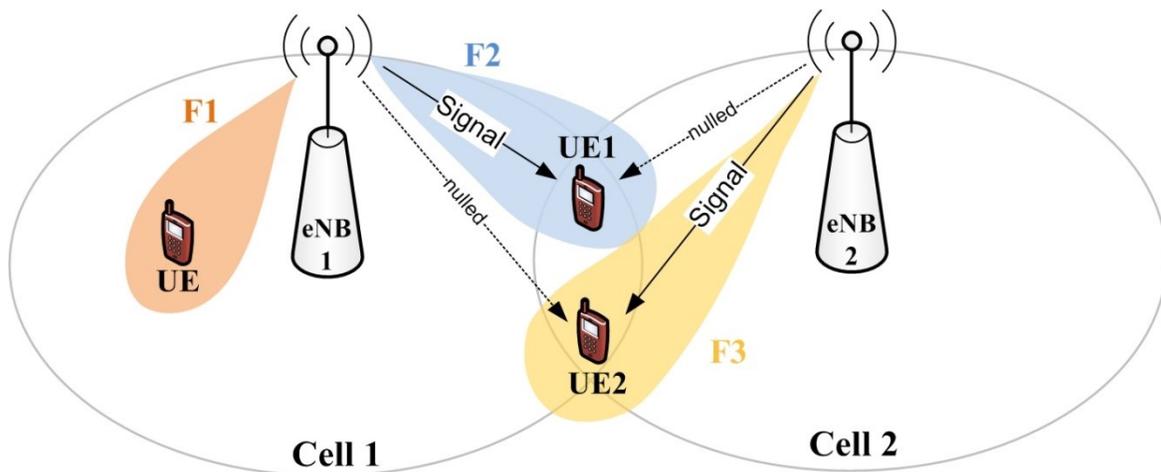
CoMP can be categorized into two categories: Coordinated Scheduling/Beamforming, and Joint Transmission, based on the way scheduling information and user data are exchanged between eNBs and transmitted to the UEs.

The main goal of Coordinated Scheduling/Beamforming (CS/CB) is to minimize interference among cell-edge UEs. In CS/CB, only control information is shared among eNBs over the backhaul, and is used to select one of the eNBs for communicating with the UE. In CS, the scheduling decisions are made in such a way that neighbouring eNBs avoid using the same time and frequency resources when communicating with UEs along the cell-edges. On the other hand, CB is a type of CoMP that avoids interference by using multi-antenna beamforming techniques to “steer” the transmission direction towards the targeted UE, while reducing the level of interference towards other UEs. In other words, CS relies on using different frequency and time resources to avoid interference, while CB relies on using different spatial resources. Figure 2.8 and Figure 2.9 illustrate a network using CS and CB respectively [7, 35].

The fundamental idea behind CS CoMP is similar to ICIC in that eNBs allocate different frequency and time resources to UEs at the edges of the cells. However, from a technical perspective, CS is a more advanced technology that requires tighter coordination between eNBs, along with more advanced signal processing techniques. In ICIC, eNBs only share information related to the interference levels within their cells. On the other hand, in CS, eNBs share scheduling information for each UE in their cell [45, 46]. Moreover, information sharing and decision making occur more frequently in CS, compared to ICIC. In essence, ICIC was designed for semi-static coordination between eNBs, while CS CoMP offers a more dynamic approach to interference mitigation [45, 47].



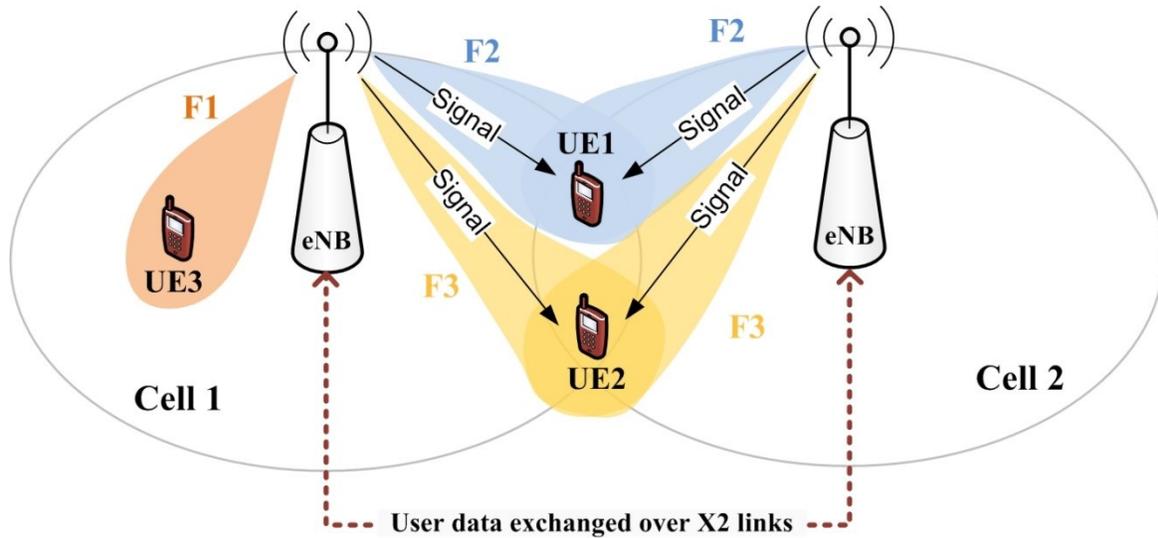
**Figure 2.8: Coordinated Scheduling CoMP.**



**Figure 2.9: Coordinated Beamforming CoMP.**

Joint Transmission (JT) CoMP allows multiple eNBs to transmit the desired signal to a UE simultaneously, effectively increasing the received signal strength at the UE side. In this case, the user data is shared between the eNBs in the coordination set along the X2 backhaul links and transmitted to the UE concurrently from multiple transmission nodes. This improves the quality of the received signal and/or cancels the effect of interference for other UEs. However, this form of coordinated transmission requires a low-latency and high-bandwidth

connection between the eNBs to support the exchange of control information, as well as user data [7]. Joint Transmission CoMP is illustrated in Figure 2.10.



**Figure 2.10: Joint Transmission CoMP.**

CoMP can also be categorized into two categories, namely centralized CoMP and distributed CoMP, based on the way information is made available at the eNBs. In the centralized architecture, a central entity is responsible for making scheduling and/or beamforming decisions based on the channel state information gathered from the UEs in the region served by the coordinated eNBs. The information is used to perform scheduling operations and signal processing and the resulting information is relayed back to the eNBs. This architecture has a high signalling overhead, caused by the additional transfer of UE channel state information and scheduling decisions to and from the centralized entity. On the other hand, a central entity is not required in the distributed approach. Instead, each UE forwards the channel state information to its serving eNB, which is responsible for forwarding it to the other eNBs in the coordination set. Each eNB receives the channel state information of not only the UEs it serves, but UEs served by the other BSs in the coordination set as well. Based on this information, each eNB can independently perform the necessary operations. This approach relies on the assumption that the schedulers in the coordinating eNBs operate in a similar manner and that the resulting scheduling decisions are identical. The distributed CoMP architecture has a reduced signalling overhead,

reduced overall infrastructure cost and less complexity compared to the centralized approach [7, 18, 36].

## 2.3 Modeling and Simulation for Cellular Networks

There are a number of simulators available for modeling and simulation (M&S) in the field of computer networks. Many of the available simulation tools are based on the discrete event simulation paradigm. These simulators model the behaviour of networks as a sequence of discrete events that occur over time. In these simulators, the nodes in the simulated network trigger events, for example, when a packet is sent between nodes. The simulator is responsible for processing these events in order, based on the scheduled event time [48].

One of the most popular discrete event simulators is NS-2 [49]. NS-2 is written in C++ and its models are based on two languages: C++ is used to define the behaviour of component models and OTcl (object-oriented extension of the Tcl language) is used for configuring network topologies and simulation parameters. The success of NS-2 can be attributed to the availability of a wide range of open-source models, allowing network researchers to simulate different network topologies without the need to define the behaviour of individual components. However, NS-2 has a relatively complex architecture, and therefore, extending the functionality of one of the open-source models or developing new models for NS-2 requires advanced skills [50].

NS-3 was released in 2008 as a successor to NS-2. Similar to NS-2, NS-3 models rely on C++ for the implementation of their behaviour. However, NS-3 no longer supports the use of OTcl for defining network topologies and simulation parameters. Instead, pure C++, and optionally Python, is used. This led to improved performance as well as scalability, at the cost of backward compatibility. Models built for NS-2 are no longer compatible with NS-3 and have to be manually ported [10].

Other simulators used in the field of network research include OPNET, OMNeT++, and JiST. In this thesis, we use Discrete Event System Specification (DEVS) to model and simulate the proposed algorithms. DEVS is presented in detail in section 2.4.

## 2.4 Discrete Event System Specification

The Discrete Event System Specification (DEVS) formalism, first introduced in 1976 by Bernard P. Zeigler [52], provides a formal framework for modelling and simulation (M&S). DEVS is hierarchical in nature, where a real system, whether discrete or continuous, can be defined as a composition of atomic and coupled models. The atomic models are the basic building blocks of the model and define the behaviour of the model. Atomic models can be interconnected to form coupled models, and a coupled model can be composed of one or more atomic or coupled models [12, 52, 53].

A DEVS atomic model is formally described by:

$$AM = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

where  $X$  is a set of input values;  $S$  is the set of sequential states;  $Y$  is the set of output values; and  $\delta_{int}$ ,  $\delta_{ext}$ ,  $\lambda$  and  $ta$  are the internal transition, external transition, output and time advance functions, respectively [11, 52].

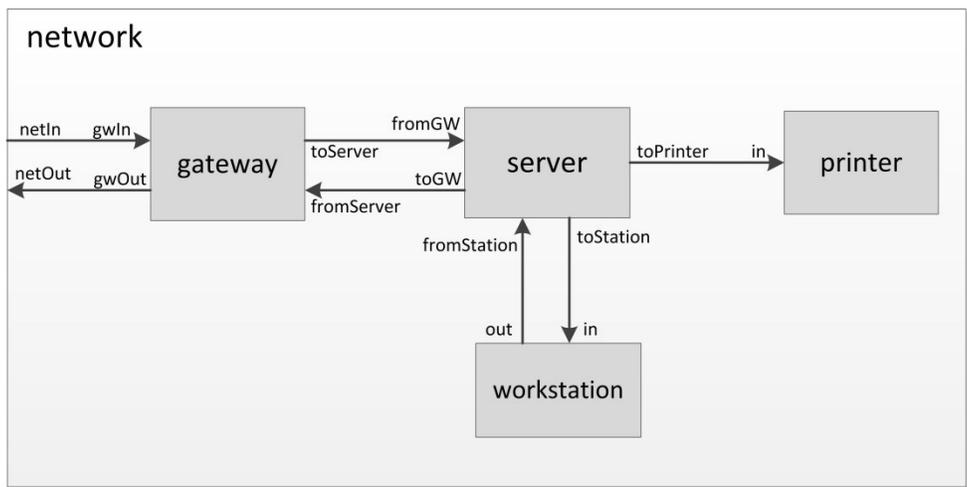
The definition of DEVS formalism allows the clear separation between the simulator and the model. This separation is important when designing models, which only requires providing an implementation of the  $\delta_{int}$ ,  $\delta_{ext}$ ,  $\lambda$ , and  $ta$  functions, without worrying about invoking them or advancing the simulation time. These tasks are carried out by the simulator. At any given point during the simulation, each DEVS model is associated with a state  $s$ , where  $s \in S$ . When the model receives an input  $x$  ( $x \in X$ ), the simulator invokes the external transition  $\delta_{ext}$  on the model and uses the time advance function  $ta$  to determine the time until the next internal transition. After that time elapses, the output function  $\lambda$  is used to obtain an output value  $y$  ( $y \in Y$ ), and the internal transition function  $\delta_{int}$  generates the new state of the model [11].

As previously mentioned, a DEVS coupled model can be composed of one or more atomic or coupled models. A coupled model is formally defined as:

$$CM = \langle X, Y, D, \{M_i\}, EIC, EOC, IC, Select \rangle$$

where  $X$  is a set of input events;  $Y$  is the set of output events;  $D$  is the set of indices that refer to the submodel components that make up the coupled model. For each  $i$  in  $D$ ,  $M_i$  is a component of the coupled model.  $EIC$  refers to the list of external input couplings, the links between the input ports of the coupled model and the submodels.  $EOC$  refers to the list of external output couplings, or the links between the submodels and the output ports of the coupled model.  $IC$  refers to the list of internal couplings, the links interconnecting the submodels. Finally, the *Select* function is used to determine the order in which the components undergo an internal transition in the event of a tie [11, 52].

For example, consider a simple network consisting of a gateway node, a router, a workstation, and a network printer. The network can be modeled using DEVS as shown in Figure 2.11. The model consists of a top-level coupled model (network) which in turn, consists of four atomic models (gateway, server, workstation, and printer). The input of the coupled model is connected to the gateway model’s “gwIn” input port. This coupling is defined in *EIC*. On the other hand, the gateway model’s “gwOut” output port is linked to the output port of the coupled model, and this link is defined in *EOC* [11]. The remaining links that interconnect the atomic models are defined in *IC*, and are shown in Figure 2.11.



**Figure 2.11: DEVS model diagram of a simple network.**

CD++ is a toolkit for developing DEVS models. Atomic models can be defined using C++ object oriented programming language. Each atomic model is implemented in a separate

class that extends the parent *Atomic* class, and provides a definition for the external, internal, output and time advance functions. Coupled models, however, are defined in a model file using a built-in specification language. This language provides the modeller with an abstract textual method to define the coupled model, along with its submodels, input and output ports, and interconnections. It also allows the modeller to specify a set of parameters for each submodel that can be used to initialize various simulation variables [11]. The model file for the networking model presented above is shown in Figure 2.12.

```
[top]
components : gateway server workstation printer
in : netIn
out : netOut
link : netIn gwIn@gateway
link : gwOut@gateway netOut
link : toServer@gateway fromGW@server
link : toGW@server fromServer@gateway
link : toStation@server in@workstation
link : out@workstation fromStation@server
link : toPrinter@server in@printer

[gateway]
in : gwIn fromServer
out : gwOut toServer

[server]
in : fromGW fromStation
out : toGW toStation toPrinter

[workstation]
in : in
out : out

[printer]
in : in
```

**Figure 2.12: Model file for a CD++ model of a simple network.**

DEVS has a number of advantages. First, the modular and hierarchical nature of DEVS allows the configuration of the model and the connections between its submodels to be easily

modified using the model file. In addition, a model based on DEVS can be integrated with other DEVS models to create models that are more complex relatively easily, without having to modify the behaviour of either model. For instance, a DEVS-based model for a cellular network can be combined with a model for crowd simulations. The output of the crowd simulation model can be used to simulate the movement of users in the LTE network model, as long as the output of the first model matches the expected input of the second model. Similarly, a vehicular traffic model can be combined with a weather forecast model to predict traffic congestions, and so on. This allows modellers from different domains to reuse existing models and integrate their work to form complex applications.

## 2.5 Related Work

There has been ongoing research in the field of CoMP and D2D in an effort to optimize these technologies. The common goal of these efforts is to increase the capacity and performance of the system, while minimizing the use of resources. CoMP research and simulation covers a wide range of issues, such as user performance, spectrum efficiency, interference management, scheduling, power management, etc. The work presented in this thesis focuses on improving the data rate available to users, especially to users near the cell edges.

In [54], the authors propose a method that allows each UE in an LTE-A network to select their transmission mode, between coherent CoMP and non-CoMP (non-cooperative transmissions) depending on a number of variables. The proposed method allows the network to exploit the benefits of CoMP while avoiding the training overhead of CoMP when it outweighs the benefits. It does this by taking into account the position of the UE and the configuration of the cooperating eNBs, as well as the overhead that would be incurred if CoMP were employed. The results showed that the proposed transmission mode-selection method achieves higher overall throughput compared to coherent CoMP for UEs located near the center of their cells. Moreover, the proposed method reduces the overhead on the backhaul network. The authors of [55] propose another Location-Aware Cooperation (LAC) method to increase the system capacity and reduce the energy consumption. This method limits the use of CoMP to users with low Signal-to-Interference-plus-Noise Ratio (SINR), which are usually located near the cell-

edges. Using this method, users near the center of their cells are served using multi-user MIMO, powered by a single eNB. The method relies on joint zero-forcing beamforming with semi-orthogonal user selection transmission in order to increase the overall system capacity and energy efficiency.

In [56], the authors use D2D techniques in a single LTE-A cell to improve the perceived data rate of a UE with poor uplink channel conditions. The method allows the UE with poor channel conditions to forward data to another UE located in close proximity with better channel conditions. The receiving UE then uploads the relayed data to its eNB as a regular upload. The authors extend this idea in [57] to allow the data to be relayed from the originating UE, through a number of UEs before being uploaded, essentially enabling the upload of data through multi-hop D2D cooperation between nearby UEs. In [58], the authors show that this approach can also lead to more efficient use of the radio spectrum as well as lower energy consumption. Compared to the standard LTE cellular approach, the results show that the proposed D2D UE cooperation method achieves about 35% reduction in the average use of radio resources and 40% reduction in energy consumption required to upload multimedia files to the network.

D2D techniques have also been used in [59] to improve the download of video files from the network. The technique, named Cached and Segmented Video Download (CSVD), splits a popular video file into pieces and selects a number of UEs (called Storage Members) in each cluster of neighbouring UEs to cache the file pieces. It then employs D2D communication to exchange these pieces between the Storage Members and other UEs in the same cluster that request the same video file. This form of cooperation increases the throughput of video transmission in cellular networks, as well as reduces the load on the network. DEVS was used to model and simulate a number of scenarios based on this technique and the results show that CSVD achieved significant improvement compared to the traditional LTE-A downlink transmission approach.

With the recent increase of deployments of small cells in heterogeneous networks, Cell Switch Off (CSO) has been introduced as a promising approach to reduce the energy consumption. This approach focuses on selecting the largest set of eNBs that can be turned off while maintaining a reasonable quality of service [60]. Traditionally, the transmit powers of

neighboring eNBs are increased in order to serve larger area. As an alternative to increasing the transmit power, the authors of [61] utilize CoMP in these scenarios to increase the received signal power. The simulation results showed that the proposed scheme achieved a more energy efficient network and increased the users' quality of service.

In [62], the authors develop an integral expression for the probability of network coverage for a typical user. The expression is then used to analyze the effect of using Joint Transmission CoMP techniques on the probability of coverage. It considers a scenario where users are randomly placed at arbitrary locations within a heterogeneous cellular network, i.e., where the network is made up of different tiers of eNBs. For a typical user at an arbitrary location, the results show that CoMP increases the user's coverage probability by 17%. In the worst case, when the CoMP cooperation is limited to only two eNBs, CoMP achieves a 24% increase in coverage probability compared to the non-cooperative case.

In this thesis, we present two algorithms (SSU and UUC) that aim to improve the uplink performance of UEs. The algorithms are based on LTE's CoMP Joint Transmission techniques to enhance the process of uploading large files to a cellular network. In addition, UUC employs LTE-A's D2D techniques to further enhance the upload process by utilizing the upload power of nearby UEs.

## Chapter 3

# Shared Segmented Upload

There are continuous efforts by mobile network operators to develop new methods to improve the performance of their networks and increase the quality of service they provide. Introducing new standards is one of the ways that network operators rely on in order to improve the performance of their networks. These standards utilize new algorithms and techniques with the end goal of increasing the overall spectrum efficiency.

In this chapter, we present a novel algorithm that aims to improve the upload performance of cell-edge UEs [14, 16, 17]. In the traditional cellular approach, the upload performance is limited by the availability of bandwidth in a single transmission link between the UE and its serving eNB, particularly for UEs near the cell edges where the received signal strength is weak. Shared Segmented Upload (SSU) aims to overcome these challenges and improve the upload performance by spreading the upload of a large file over multiple transmission links to the eNBs in the CoMP coordination set. Moreover, in this chapter, we present a simple DEVS model to model and simulate an LTE network. The model was used to access the performance of SSU, compared to the traditional cellular approach.

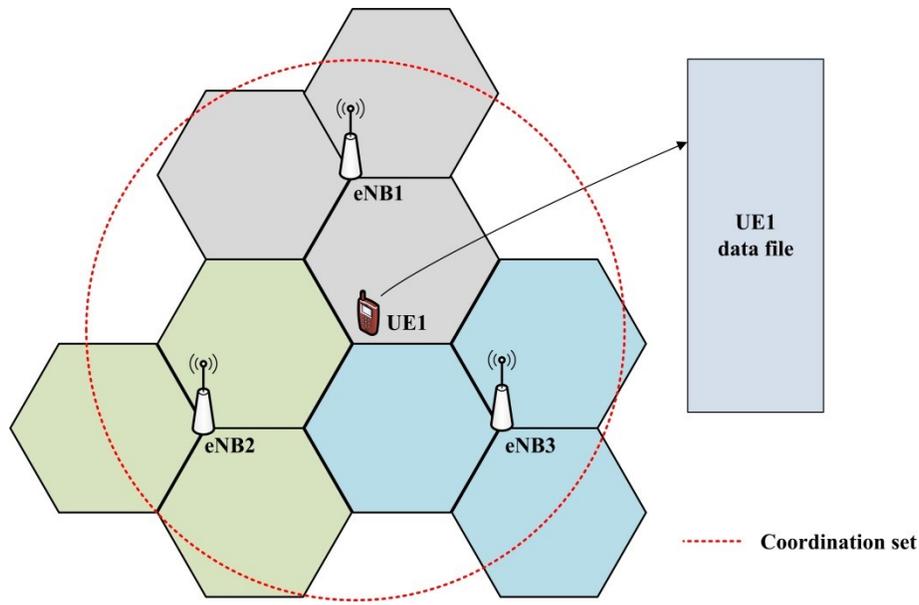
### 3.1 Algorithm Overview

As previously mentioned, cell-edge users suffer from low received signal power as well as a higher level of interference from neighboring cells. SSU aims to improve the quality of

service for a typical UE near the cell edges by allowing it to upload a large file to multiple eNBs in a distributed CoMP architecture, simultaneously. SSU has a number of common points with BitTorrent [15]. BitTorrent is a peer-to-peer protocol used to speed up the sharing of large files on the Internet by allowing the users to join a swarm of hosts to download and upload from each other, simultaneously. BitTorrent relies on segmented file transfer where the file being distributed is divided into a large number of segments, called pieces. The upload begins by one user that starts sending pieces to other users, one at a time. When another user successfully receives a piece of the file, the user can become a source of that piece for other users. BitTorrent is considered as an alternative to the single source, multiple mirror sources technique for distributing data [15]. This enables BitTorrent to perform efficiently even over networks with low bandwidth.

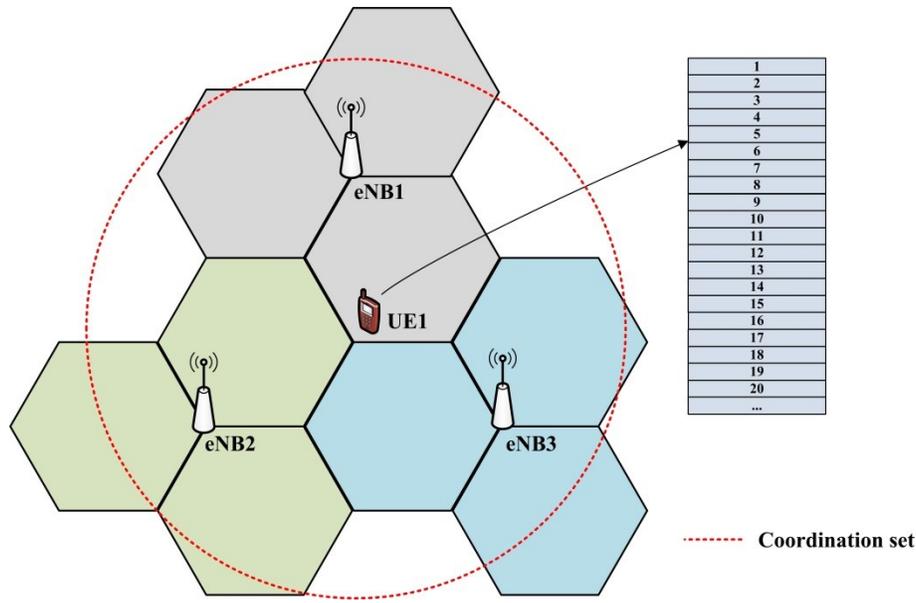
SSU adapts this technique to improve the data upload of a large file from a single UE to a set of eNBs, by transferring the file in small segments, called pieces. This technique can solve the bottlenecks caused, for instance, by UEs uploading large data files to the network, improving the upload performance and the overall quality of experience. In SSU, the pieces are transmitted by the UE to the eNBs independently. Therefore, the eNBs with better transmission links to the UE will receive more pieces of the original data file. This process speeds up the data upload, especially for UEs located near the cell borders. Finally, the pieces received by the eNBs in the coordination set are gathered at the UE's serving eNB and are used to reconstruct the original data file, like the pieces of a puzzle.

Consider the example shown in Figure 3.1. UE1, located near the border, wishes to upload a large file using SSU. The UE is able to communicate with three eNBs, namely eNB1, eNB2, and eNB3, which form a CoMP coordination set to support the UE's file upload. In this example, let us assume that eNB1 is UE1's serving eNB.



**Figure 3.1: A UE (UE1) wishes to upload a data file to the network.**

Prior to the beginning of the upload process, the data file that the UE wishes to upload is fragmented into a number of pieces (Figure 3.2). The size of the pieces is selected based on the size of the data file as well as a number of other factors, and is typically a power of two. There is a trade-off between the piece size and the effectiveness of SSU. A large piece would make SSU less effective, as it becomes similar to uploading a large file using the traditional approach. On the other hand, a small piece would increase the signalling overhead of SSU. The optimal piece size depends on a number of factors, including the original data file size, the number of eNBs participating in the CoMP coordination set, the quality of the transmission links, and the speed of the UE. In the BitTorrent protocol, the most common piece sizes are 256KB, 512KB and 1MB [15]. In SSU, all the file pieces are of equal length, except for the last piece, whose size can be equal to or smaller than the remaining pieces. The UE then creates a file descriptor, called the *MetaInfo* file, which contains information about the data file. Compared to the data file, the *MetaInfo* file is relatively small and can be quickly transferred to the UE's serving eNB.

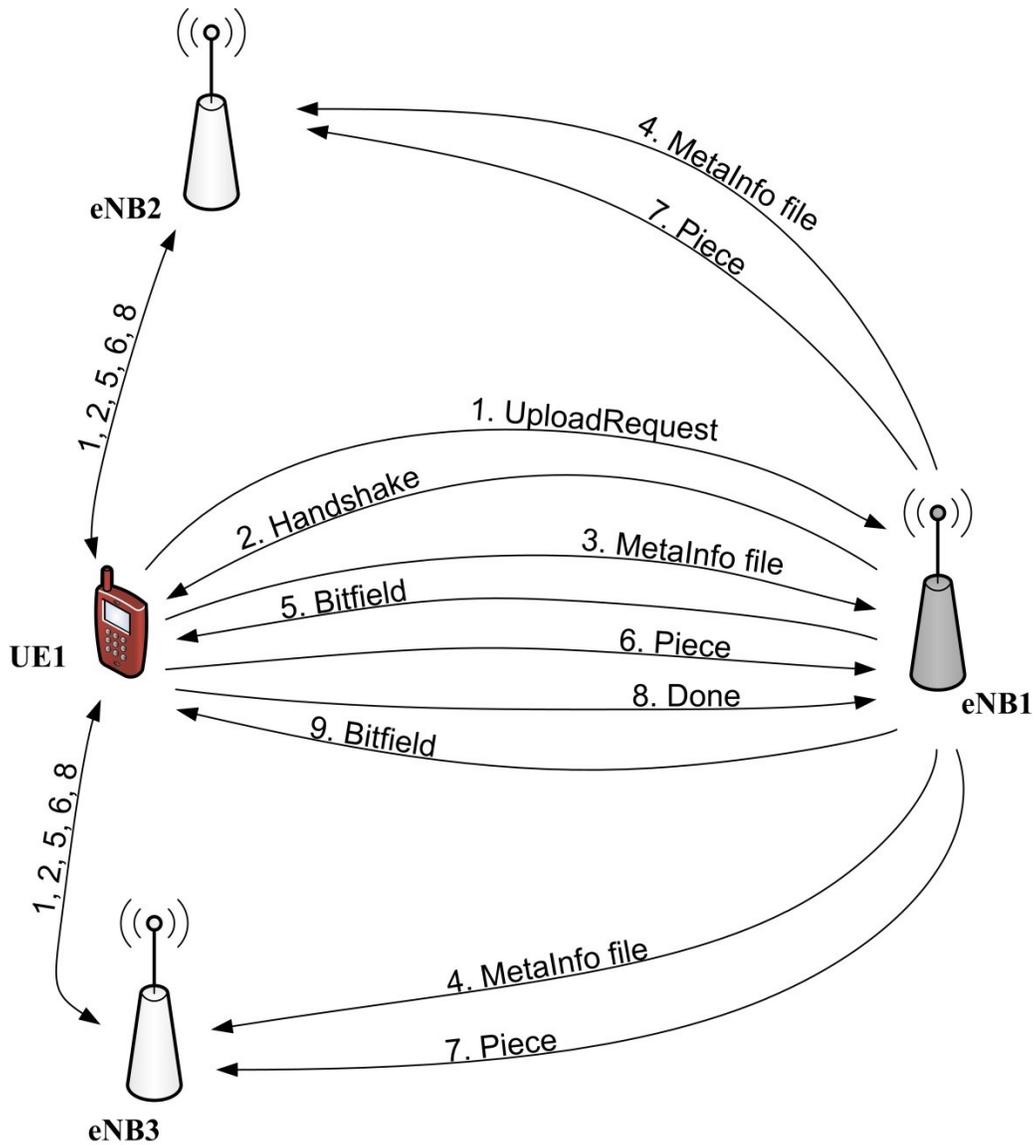


**Figure 3.2: UE1 divides the data file into a number of pieces.**

The upload process begins with the UE sending an *UploadRequest* message to the eNBs in the CoMP coordination set. The non-serving eNBs can choose to join the coordination set, and do so by sending a *Handshake* message to the UE. This decision typically depends on the quality of the transmission links between the UE and eNB, as well as the traffic load on the eNB. Upon receiving the *Handshake* message from the eNBs, the UE sends the *MetaInfo* file to the serving eNB. The serving eNB is responsible for sharing the *MetaInfo* file with other non-serving eNBs of the coordination set, using the X2 backhaul links. The serving eNB acknowledges the reception of the *MetaInfo* file by sending a *Bitfield* message to the UE. The *Bitfield* message also informs the UE of any pieces that have been previously uploaded and are available at the eNBs. The UE starts uploading pieces of the file to eNBs in the coordination set, simultaneously. The pieces that are received by the non-serving eNBs are forwarded to the serving eNB over the X2 backhaul links. After all the pieces have been uploaded, the UE sends a *Done* message to all the eNBs in the coordination set. Upon receiving the *Done* message, the serving eNB sends another *Bitfield* message to the UE. This *Bitfield* message lists the reception status of the pieces and allows the UE to retransmit any missing pieces. The retransmitted pieces are followed by another *Done* message. The serving eNB sends the UE an updated *Bitfield* message after receiving the *Done* message. If all the pieces of the file have been received correctly by the serving eNB, the

upload is completed. Otherwise, the upload process is restarted. Figure 3.3 illustrates the steps of the upload process, which are summarized below.

1. The UE sends an *UploadRequest* message to each eNB in the CoMP coordination set.
2. The eNBs reply using a *Handshake* message.
3. The UE sends the *MetaInfo* file that describes the original data file, to the serving eNB.
4. The serving eNB forwards the *MetaInfo* file to the non-serving eNBs in the CoMP coordination set.
5. The eNBs confirm the receipt of the *MetaInfo* file by sending a *Bitfield* message to the UE. The *Bitfield* message informs the UE of any pieces that are already available from previous uploads.
6. The UE uploads the file in pieces to the eNBs in the CoMP coordination set, simultaneously.
7. Upon receiving a *Piece* message, the non-serving eNBs forwards it to the serving eNB.
8. Following the last *Piece* message, the UE sends a *Done* message to all the eNBs in the CoMP coordination set.
9. The serving eNB confirms the receipt of the pieces using a *Bitfield* message. If all the pieces have been received by the network, the upload is complete. However, if one or more pieces were not received by the eNBs, the UE retransmits the missing pieces by repeating steps 6 to 9. If, after the retransmission, one or more pieces were not received by the network, the upload process is terminated.



**Figure 3.3: The steps of a Shared Segmented Upload process.**

## 3.2 SSU Messaging Structure

The following subsections discuss the details of the messages exchanged between UEs and eNBs employing SSU. The first three fields are common to all the messages. The *messageType* field, as the name suggests, specifies the type of message being transmitted. The second and third fields identify the ID of the sender and receiver of each message, respectively. Depending on the type of message, the sender and receiver IDs can represent a UE or an eNB.

### 3.2.1 UploadRequest message

The upload process begins with the UE sending *UploadRequest* messages to all the eNBs within range of the UE. The message contains a field that identifies the UE's serving eNB. This information is used by any non-serving eNBs that wish to communicate with the serving eNB prior to the upload. The message also contains basic information that identifies the file, including the file size. This information is used by the non-serving eNBs to decide whether to join the CoMP coordination set. The structure of the *UploadRequest* message is shown in Table 3.1

| # | Field             | Type |
|---|-------------------|------|
| 1 | messageType       | int  |
| 2 | sender ID         | int  |
| 3 | receiver ID       | int  |
| 4 | serving eNB ID    | int  |
| 5 | file ID           | long |
| 6 | file size (bytes) | long |

**Table 3.1: *UploadRequest* message structure.**

### 3.2.2 Handshake message

If the non-serving eNBs that receive an *UploadRequest* message decide to join the coordination set for that UE's file upload, they respond with a *Handshake* message. The message informs the UE of the eNB's decision. The message structure is shown in Table 3.2.

| # | Field       | Type |
|---|-------------|------|
| 1 | messageType | int  |
| 2 | sender ID   | int  |
| 3 | receiver ID | int  |
| 4 | file ID     | long |

**Table 3.2: *Handshake* message structure.**

### 3.2.3 MetaInfo file message

Prior to the upload process, the UE creates a small data file descriptor. The *MetaInfo* file identifies the original data file and contains information about the pieces. This information is required by the serving eNB at the end of the upload process to identify the missing file pieces, and to reconstruct the original data file. Moreover, the information is used by the non-serving eNBs to identify the received pieces and forward them to the serving eNB. Each piece is identified using an SHA-1 hash code. The hash codes are 20-byte long cryptographic strings, and are concatenated together to form the “*piece hash values*” field in the *MetaInfo* file. Therefore, the length of that field depends on the number of pieces. The SHA-1 hash codes are used in the upper layers of the LTE protocol stack to validate the received pieces and detect any transmission errors. The remaining fields are similar those in the previous messages. The structure of the *MetaInfo* file is shown in Table 3.3.

| # | Field              | Type   |
|---|--------------------|--------|
| 1 | messageType        | int    |
| 2 | sender ID          | int    |
| 3 | receiver ID        | int    |
| 4 | file ID            | long   |
| 5 | file size (bytes)  | long   |
| 6 | piece size (bytes) | long   |
| 7 | piece hash values  | string |

**Table 3.3: *MetaInfo* file message structure.**

### 3.2.4 Bitfield message

When the eNBs receive the *MetaInfo* file, either directly from the UE, or through the serving eNB, they respond with a *Bitfield* message. The “*pieces*” field is used to list the IDs of any pieces that are available at any of the eNBs from previous uploads. The length of the field, and therefore the message, depends on the number of pieces that have been previously uploaded. Table 3.4 shows the structure of the *Bitfield* message.

| # | Field       | Type        |
|---|-------------|-------------|
| 1 | messageType | int         |
| 2 | sender ID   | int         |
| 3 | receiver ID | int         |
| 4 | file ID     | long        |
| 5 | pieces      | array[long] |

**Table 3.4: *Bitfield* message structure.**

### 3.2.5 Piece message

*Piece* messages are used to transmit portions of the data file. When a *Piece* message is received by the serving eNB, the “*payload*” is used to generate an SHA-1 hash code, which is then compared to the corresponding hash code in the *MetaInfo* file to ensure that the data has been received without transmission errors. The structure of a *Piece* message is shown in Table 3.5.

| # | Field       | Type |
|---|-------------|------|
| 1 | messageType | int  |
| 2 | sender ID   | int  |
| 3 | receiver ID | int  |
| 4 | file ID     | long |
| 5 | piece ID    | long |
| 6 | payload     |      |

**Table 3.5: Structure of a *Piece* message.**

### 3.2.6 Done message

After the last piece has been uploaded by the UE, the UE sends a *Done* message to indicate the end of the upload process. The structure of the *Done* message is shown in Table 3.6. The “*last piece ID*” refers to the ID of the last piece of the data file.

| # | Field         | Type |
|---|---------------|------|
| 1 | messageType   | int  |
| 2 | sender ID     | int  |
| 3 | receiver ID   | int  |
| 4 | file ID       | long |
| 5 | last piece ID | long |

**Table 3.6: Done message structure.**

### 3.3 Modeling of SSU using DEVS

To study the performance of SSU, a DEVS model for an LTE mobile network that implements SSU was developed. As seen in Figure 3.4, the model consists of various sub-coupled models. The top-level coupled model is called *Area*. This coupled model contains three other coupled models, namely *Atmosphere*, *UEManager*, and *BSManager*, as well as one atomic model, called *LogManager*. The *LogManager* atomic model is responsible for gathering statistical data from the other models, and output them into log files during the simulations. The *Atmosphere* coupled model is used to model the communication between each pair of eNBs (or BSs) and UEs. Instead of defining interconnections between each pair of UE/BS coupled models (which can grow quickly as the size of the simulated network increases), the *Atmosphere* model is used to receive all the outgoing messages from UEs and BSs, and broadcast them to all the other models. The *BS* and *UE* models can then filter the incoming messages, based on their IDs and the destination address fields of the received messages. Both the *UEManager* and *BSManager* coupled models have similar structures, each having a registry unit (a DEVS atomic model). The registry units are responsible for a number of management and control functions of the corresponding UEs and BSs. In addition to the registry units, *BSManager* and *UEManager* can contain any number of *BS* or *UE* coupled models, as defined in the network topology and the simulation scenario. The *BS* and *UE* coupled models are each composed of two atomic models, *Queue* and *Processor*. The *Queue*, as the name suggests, is responsible for queuing the incoming messages and forwarding them to the *Processor* atomic model on a FIFO (first-in-first-out) basis. The *Processor* atomic models implement the behaviour of UEs or eNBs employing SSU.

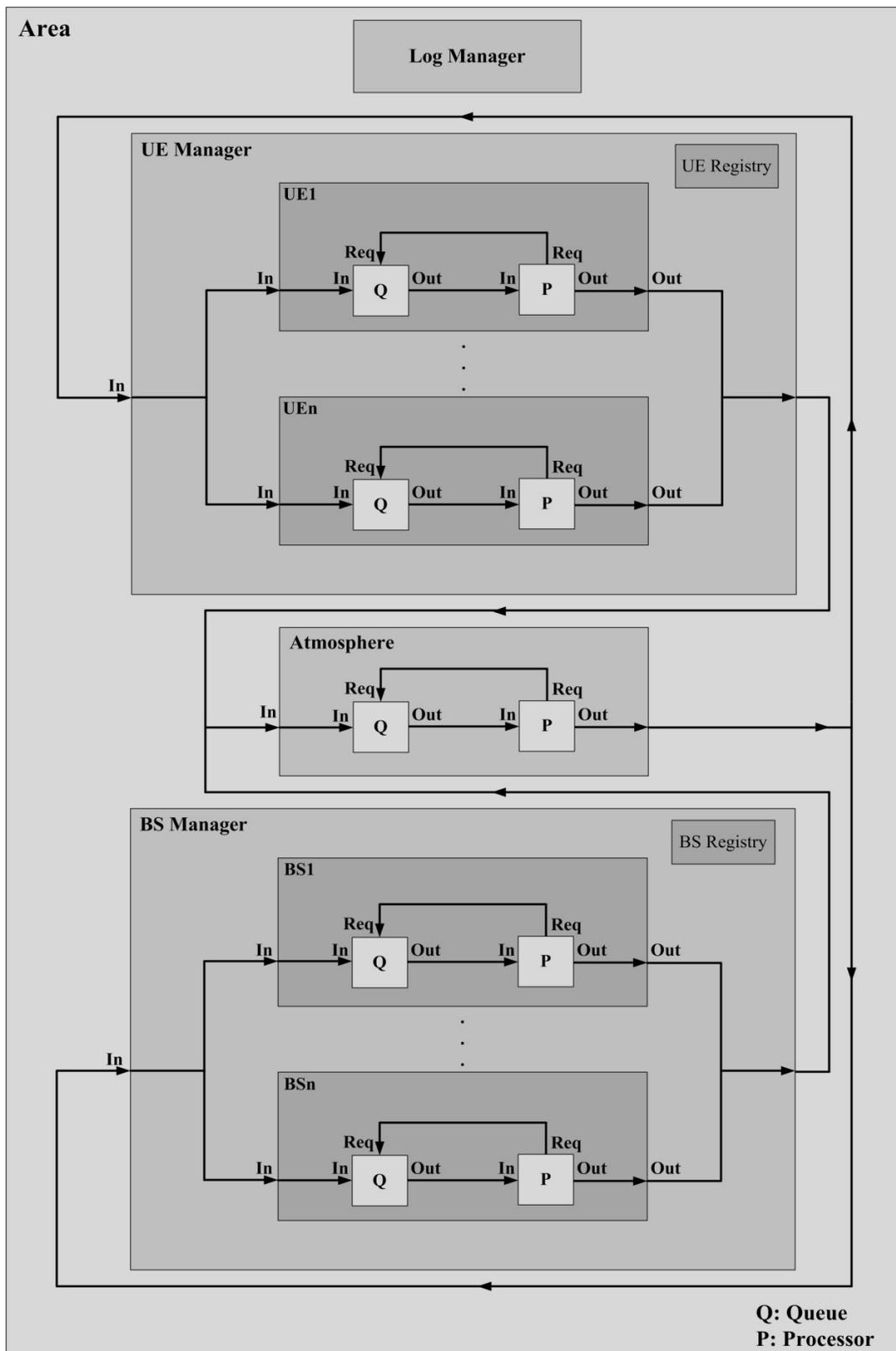


Figure 3.4: DEVS model diagram for the mobile network model.

As mentioned earlier, the structure of DEVS models is defined in Model files. A Model file is used to define DEVS coupled models, their input and output ports, component models, as well as the interconnections between them. The file is also used to define a list of parameters for each atomic model. These parameters are read by the corresponding atomic models during the initialization of the simulation. Figure 3.5 shows a simplified segment of the Model file that defines the model presented in Figure 3.4. The complete model file is available online at [63].

**[top]**

components : logManager@LogManager atmosphere UEManager BSManager

...

link : out@ atmosphere in@UEmanager

link : out@ atmosphere in@BSmanager

link : out@UEmanager in@atmosphere

link : out@BSmanager in@ atmosphere

**[logManager]**

areaConfiguration : rural

bsCounter : 16

ueCounter : 64

...

**[UEManager]**

components : UERegistry@UERegistry UE1 UE2 ... UE64

...

**[UE1]**

components : UE1Queue@UEQueue UE1Processor@Node

...

**[UE1Queue]**

UEId : 1

**[UE1Processor]**

UEId : 1

currentX : 23729

currentY : 14210

...

**[BSmanager]**

components : BSregistry@BSRegistry BS1 ... BS16

...

```

[BS1]
components : BS1Queue@BSQueue BS1Processor@BSProcessor
...

[BS1Queue]
...

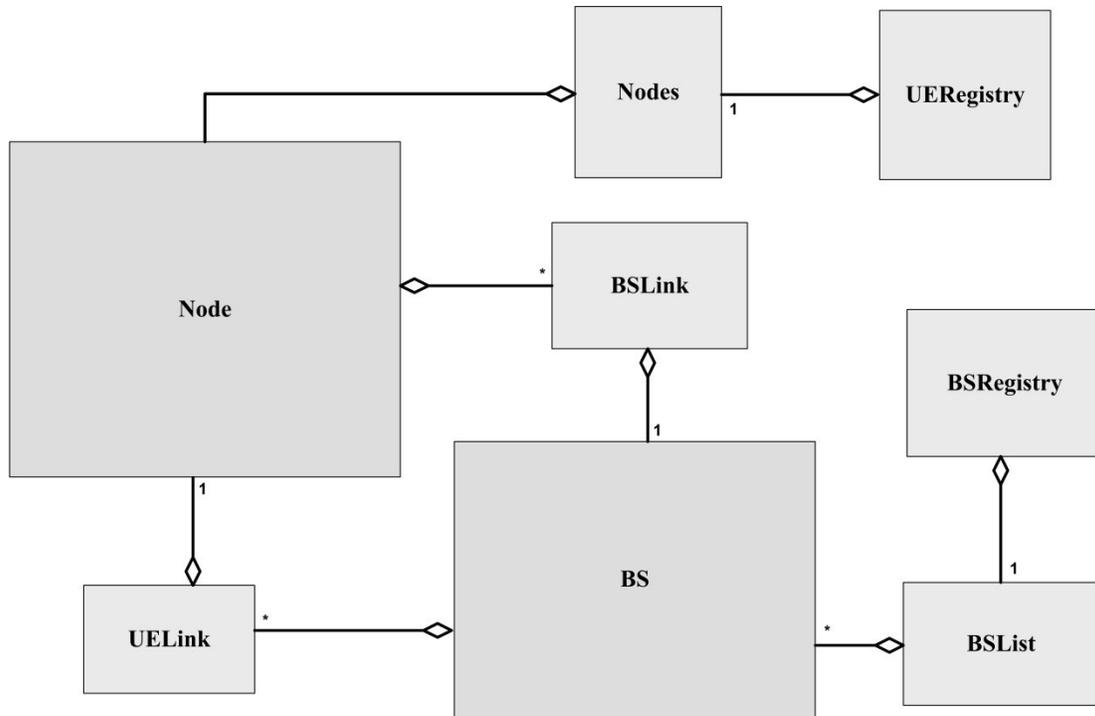
[BS1Processor]
BSId : 1
areaConfiguration : 0
Dhb : 15
Hb : 45
frequency : 900
...

```

**Figure 3.5: Simplified segment of the Model file that defines the SSU DEVS model.**

As seen in Figure 3.5, the *Area* coupled model is defined as the top model using the *[top]* clause. The components of the *Area* model (all the second level models) are defined using the *components* tag. Here, the atomic component models are named and defined as instances of their classes, using the format *instanceName@modelName*, while coupled models are simply named and defined in a separate group in the Model file. For instance, in Figure 3.5, *logManager* is an instance of the atomic model class *LogManager*, while *UEManager* is a name given to a coupled model, which is defined later in the configuration file. This allows a coupled model to be composed of multiple instances of the same atomic model definition. The *link* tag defines an interconnection between two components, or between a component and an input/output port in coupled model itself. Instances of the atomic models are listed using the *[atomicModelInstanceName]* clause, and define the list of parameters that belong to that atomic model instance. For example, *logManager* lists a number of user-defined configuration parameters, such as *areaConfiguration*, *bsCounter*, and so on.

Each of the atomic models is defined in a separate C++ class. Aside from these atomic models, a number of other passive classes have been added to complete the model. Figure 3.6 shows a simplified, high-level Unified Modelling Language (UML) class diagram of the model classes.



**Figure 3.6: Simplified segment of the Model file that defines the SSU DEVS model.**

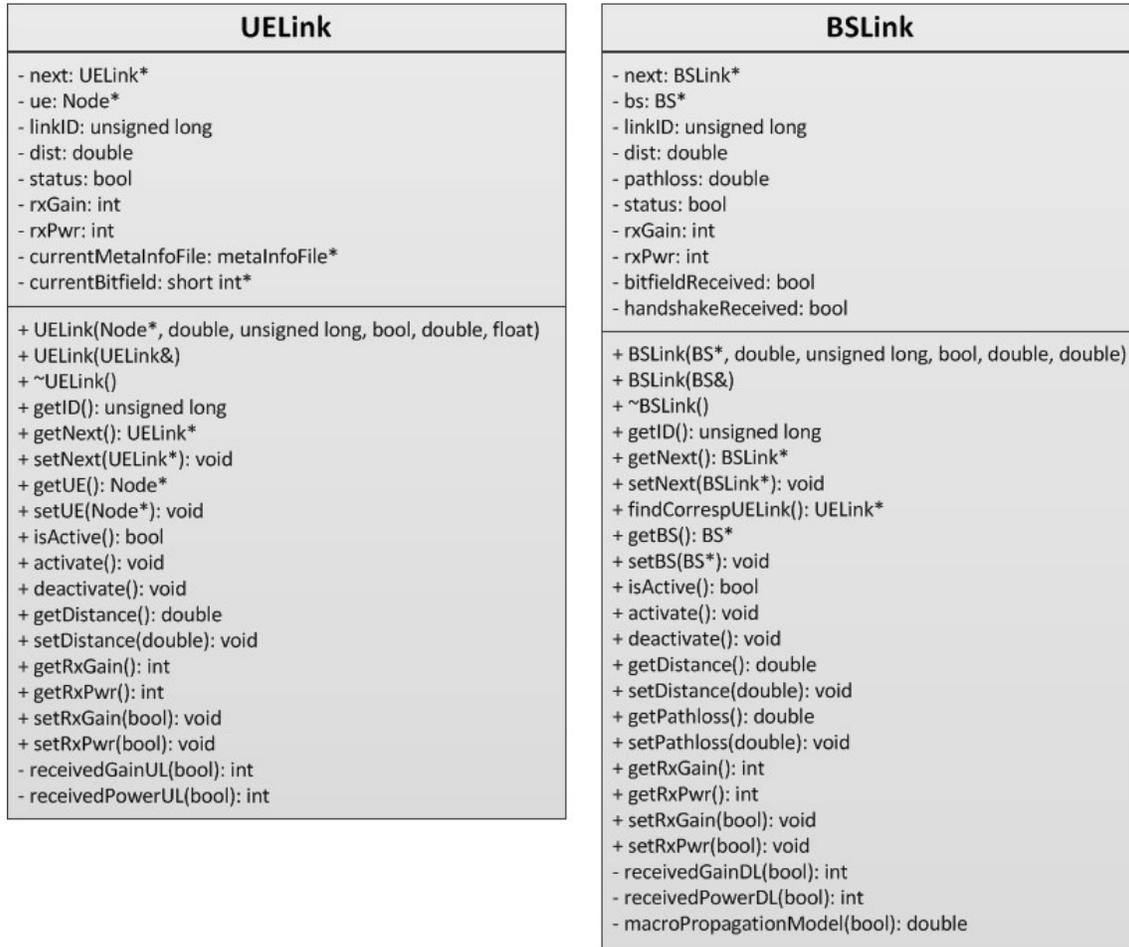
The *BS* class (shown in detail in Figure 3.7a) represents a *BSProcessor* (from the model shown in Figure 3.4) using the following parameters: id, type, coordinates, height from the ground, height from the average roof top, carrier frequency, transmission power, antenna gain, along with a list of connections to the UEs within range. Similarly, the *Node* class represents the behaviour of a *UEProcessor* using the following parameters: id, current conditions, destination coordinates, speed, transmission power, antenna gain, as well as a list of connections to the BSs within range. A detailed class diagram for the *Node* class is shown in Figure 3.7a. These parameters are initialized independently through the Model file during the initialization of the simulation. The *BS* and *Node* classes contain functions that define the behaviour of the corresponding component using the algorithm being modeled.

Each instance of the *UELink* class defines a single downlink transmission link between a BS (as a source) and a UE (as a destination). The class contains the downlink transmission parameters, including separation distance, path loss, received power and data rate. Similarly, the *BSLink* class defines an uplink transmission link between a UE and BS. The uplink transmission parameters are similar to those defined in the *UELink* class. Based on the propagation model,

both classes have functions to calculate the corresponding transmission link parameters, such as path loss and received power. Each instance of the *Node* and *BS* classes hold a pointer to the head of linked lists of *BSLink* and *UELink* respectively. The detailed class diagrams for *UELink* and *BSLink* are shown in Figure 3.7b.



(a)

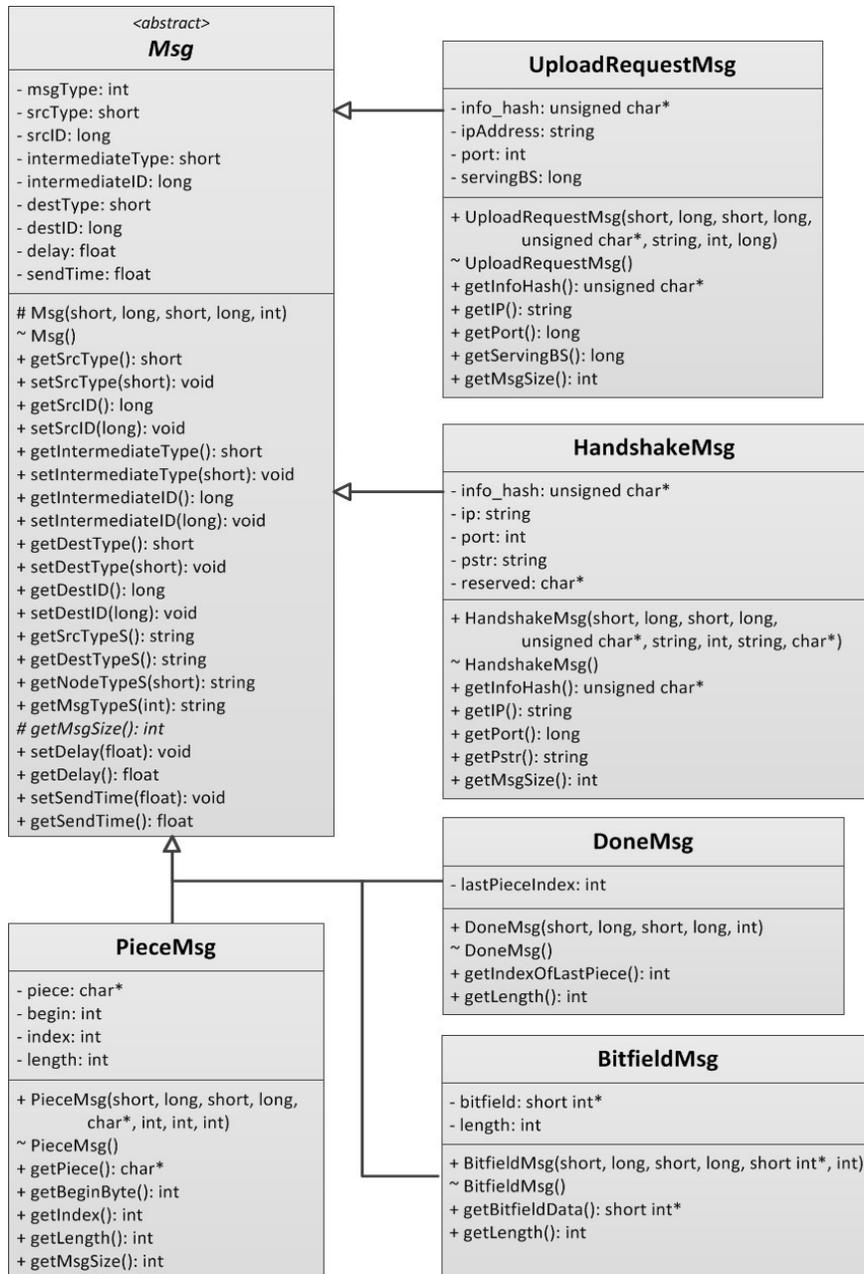


(b)

**Figure 3.7: Detailed UML class diagrams.**

*UERegistry* and *BSRegistry* are atomic models that are used to initialize the lists of UEs and BSs, respectively. The *UERegistry* class holds a reference to an instance of *Nodes*, a class that is responsible for maintaining the list of UEs. The *UERegistry* class periodically triggers the *Nodes* class to update the positions of the UEs, based on their current positions, destinations, and current speeds. Moreover, the *UERegistry* class periodically updates the status of the communication channels between UEs and BSs (as defined in *UeLink* and *BSLink*). The channel status includes the validity of the links, depending on the distance between the corresponding UE and BS, as well as the uplink and downlink transmission parameters discussed earlier. Likewise, the *BSRegistry* holds a reference to an instance of *BSList*, which is responsible for maintaining the list of BSs during the simulation.

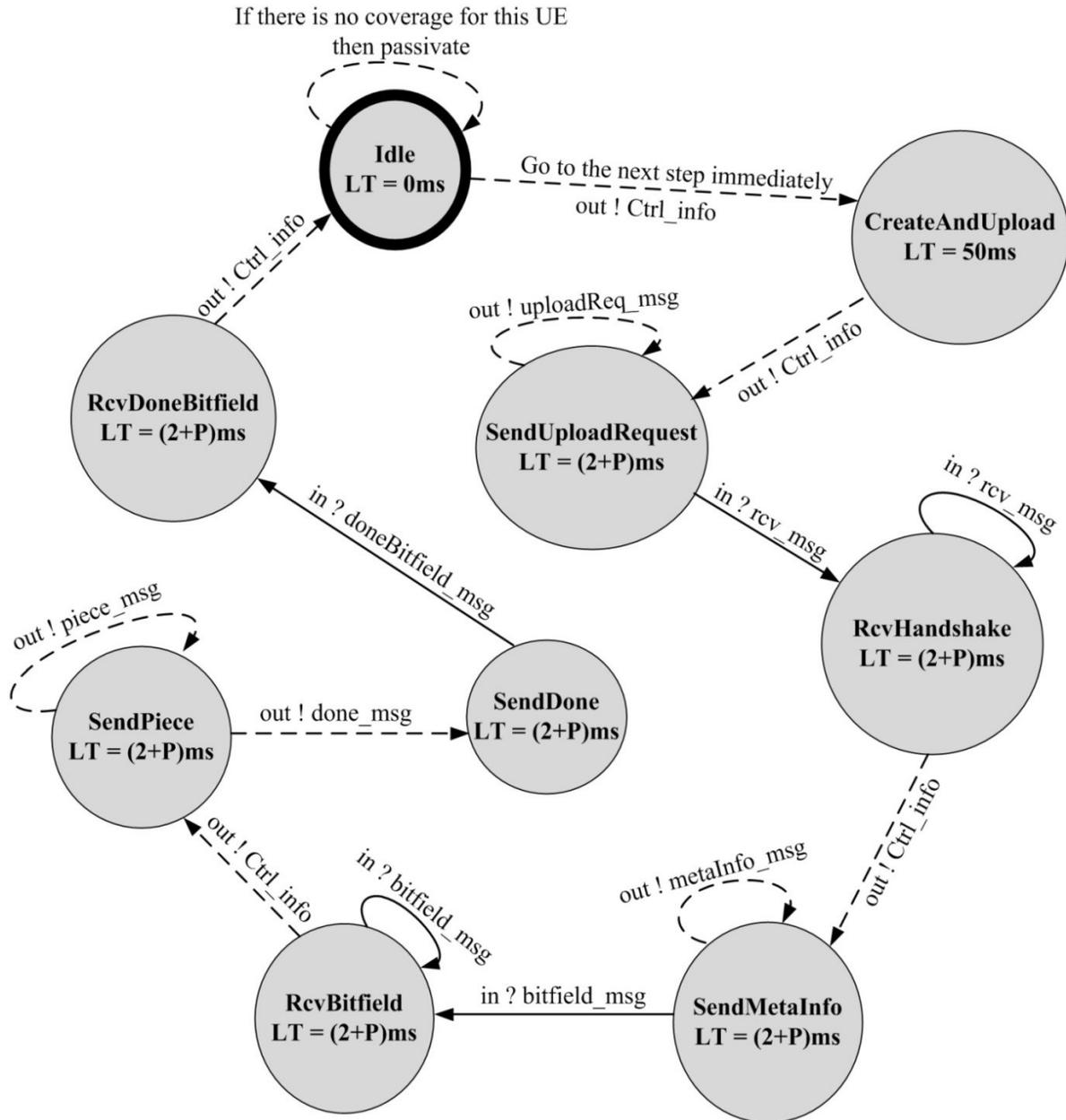
A class hierarchy for messages has been implemented to encapsulate the contents of the messages used by SSU and allow the UE and BS models to communicate with fewer messages. The *Msg* class is an abstract superclass that represents any message sent by a UE/BS. This class includes fields that are common to all SSU messages, such as message type, and source and destination addresses. Each of the messages defined in SSU is implemented as a subclass of the parent *Msg* class. Figure 3.8 shows a UML class diagram for the messaging class hierarchy.



**Figure 3.8: UML class diagram for the messaging class hierarchy.**

In the SSU DEVS model, each UE is limited to the upload of only one file at a time. Therefore, the *UEProcessor* state transitions directly correspond to the different steps of the proposed algorithm. The *UEProcessor* states are *Idle*, *CreateAndUpload*, *SendUploadRequest*, *RcvHandshake*, *SendMetaInfo*, *RcvBitfield*, *SendPiece*, *SendDone*, and *RcvDoneBitfield*. A DEVS graph for *UEProcessor* is shown in Figure 3.9. A DEVS graph defines the behaviour of an atomic model in an abstract manner. In a DEVS graph, the initial state is denoted with a thick solid outline. State transitions are defined according to the model's definition of the DEVS internal and external transition functions. The state transitions that correspond to an internal transition are represented using dashed arrows that connect the source and destination states. Each internal transition can also be associated with a port/value pair that corresponds to the output function. An internal transition that triggers an output is represented by " $p ! v$ ", where  $p$  is the output port and  $v$  is the output value. Similarly, an external transition triggered by an input is denoted using a solid arrow that connects the two states. The transition is denoted using " $p ? v$ ", where  $p$  is the input port and  $v$  is the input value that is expected for the external transition to occur.

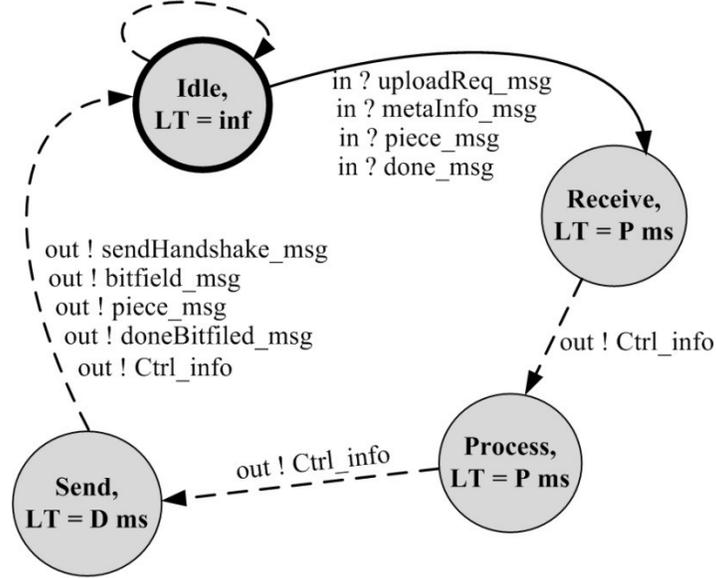
On the other hand, in the SSU DEVS model, a BS is able to support the upload of multiple files originating from several UEs, as well as control and data messages from neighboring BSs. To allow this, the *BSProcessor* was designed to handle any number of communication sessions with other model components. To simplify its state transitions, *BSProcessor* only cycles through four states, namely, *Idle*, *Receive*, *Process*, *Send*, once for each message received. The DEVS graph for *BSProcessor* is shown in Figure 3.10. The state of the upload session that belongs to a received message is stored in the *UELink* instance that corresponds to the UE that sent the message. The *UELink* states are *Idle*, *RcvUploadRequest*, *SendHandshake*, *RcvMetaInfo*, *SendMetaInfo*, *RcvPiece*, *SendPiece*, *RcvDone*, and *SendBitfield*. A state diagram that illustrates these states and the transitions between them is shown in Figure 3.11.



**P:** Additional required Process Time, based on the message size in ms

**Figure 3.9: UEProcessor DEVS graph.**

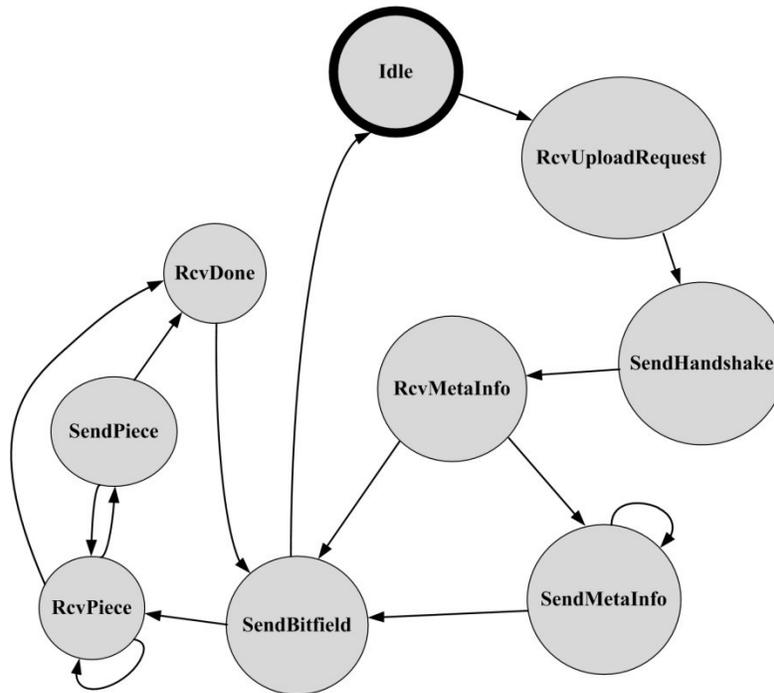
After initialization, If there is no UE in the BS coverage area, then it goes to passivate mode



**P:** Process Time, in ms

**D:** Transmission delay (based on the available data rate and the message size), in ms

**Figure 3.10: BSProcessor DEVS graph.**



**Figure 3.11: UELink state diagram.**

## 3.4 Evaluation of SSU

### 3.4.1 Non-cooperative Algorithm

In order to evaluate the performance of SSU, another conventional non-cooperative algorithm was implemented. Unlike SSU, the non-cooperative algorithm does not involve a dynamic cooperation between eNBs. Instead, in the non-cooperative algorithm, a UE only interacts with its serving eNB. In this algorithm, the upload begins with a request message, similar to the *UploadRequest* message used in SSU. Once the UE's serving eNB acknowledges the request, the UE can begin uploading the data file. The upload occurs in a series of variable sized packets. The size of the packets depends on the separation distance between the UE and the serving eNB, and the available bandwidth. After the last data packet has been uploaded, the UE sends the serving eNB a *Done* message to indicate that the file has been uploaded.

### 3.4.2 Propagation model and simulation parameters

As previously mentioned, the *UERegistry* atomic model periodically triggers its instance of the *Nodes* class to update the UEs' locations, based on their current locations, their destinations, as well as their current speeds. Moreover, the *UERegistry* model periodically updates the transmission parameters based on the propagation model for the links between each pair of eNBs and UEs. The transmission parameters include the propagation path loss, noise, received signal power and the available data rate. The available data rate is then used to determine the amount of data that can be sent in a transport block.

For urban and rural areas, our SSU DEVS model uses a propagation model defined in [64]. The model is based on the Okumura-Hata propagation model for urban areas [65]. The macro cell propagation losses for urban and rural areas are calculated using equation (1) and (2) respectively [64].

$$L_{urban} = (40 * (1 - (4 * 10^{-3} * Dhb)) * \log_{10} R) - (18 * \log_{10} Dhb) + (21 * \log_{10} f) + 80dB \quad (1)$$

$$L_{rural} = 69.55 + (26.16 \times \log_{10} f) - (13.82 \times \log_{10} hb) + \left( (44.9 - (6.55 \times \log_{10} hb)) \times \log_{10} R \right) - (4.78 \times (\log_{10} f)^2) + (18.33 \times \log_{10} f) - 40.94 \quad (2)$$

where  $R$  is the separation distance between the eNB and UE (in kilometers),  $f$  is the carrier frequency (in MHz),  $Dhb$  is the eNB antenna height measured from the average rooftop level (in meters), and  $Hb$  is the antenna height above ground (in meters).

The path loss can then be calculated using equation (3), where  $LogF$  is the log-normally distributed shadowing [64].

$$pathlossPlusShadowing = L + LogF \quad (3)$$

The received signal power at a UE or eNB is then calculated using equation (4), where  $Pwr_R$  is the received signal power,  $G_R$  is the antenna gain of the receiver,  $Pwr_T$  is the transmitter signal power,  $G_T$  is the antenna gain of the transmitter, and  $MCL$  is the minimum coupling loss (in dB).

$$Pwr_R = Pwr_T - \text{Max}(pathlossPlusShadowing - G_T - G_R, MCL) \quad (4)$$

Finally, the data rate of the given transmission link can be calculated taking into account additive white Gaussian noise, using equation (5), where  $B$  is the channel bandwidth (in MHz), and  $N_0$  is the noise variance.

$$data\ rate = B \log_2 \left( 1 + \frac{Pwr_R}{N_0 \times B} \right) \quad (5)$$

The simulations were carried out in both urban and rural area settings. In the urban area setting, two operating frequencies are considered: 900 MHz and 2000 MHz, while the operating frequency used in rural areas is 900 MHz. In both settings, 5 MHz was used as the transmission channel bandwidth. In addition, a fixed noise density (-174 dBm/Hz) was used, while the log-normally distributed shadowing (LogF) was set to a standard deviation of 10 dB. The maximum eNB and UE transmission powers were set to 43 dBm and 21 dBm respectively. The simulation/transmission parameters that are common to both area settings are summarized in Table 3.7. Table 3.8 and Table 3.9 list the parameters specific to urban and rural areas, respectively.

| Parameter                             | Value           |
|---------------------------------------|-----------------|
| Transmission bandwidth                | 5 MHz           |
| Noise Density                         | -174 dBm/Hz     |
| Dhb                                   | 15 meters       |
| Hb                                    | 45 meters       |
| LogF standard deviation               | 10 dB           |
| Data file size (to be uploaded by UE) | 0.5 MB to 64 MB |
| Maximum eNB power                     | 43 dBm          |
| Maximum power per DL channel          | 30 dBm          |
| Minimum eNB power per UE              | 15 dBm          |
| Noise figure (eNB)                    | 5 dB            |
| Maximum UE transmission power         | 21 dBm          |
| Minimum UE transmission power         | -50 dBm         |
| Noise figure (UE)                     | 9 dB            |

**Table 3.7: Common simulation/transmission parameters.**

| Parameter          | Value   |          |
|--------------------|---------|----------|
| Carrier frequency  | 900 MHz | 2000 MHz |
| Antenna gain (eNB) | 12 dB   | 15 dB    |
| MCL                | 70 dB   |          |

**Table 3.8: Simulation/transmission parameters specific for urban areas.**

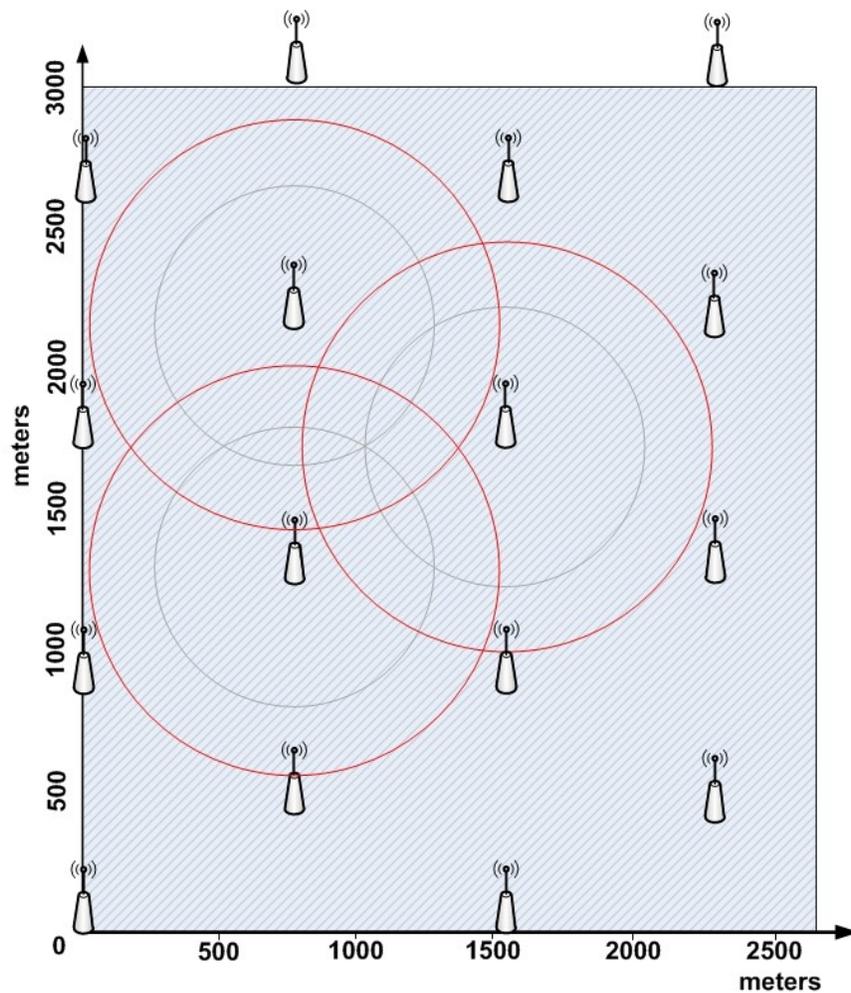
| Parameter          | Value    |
|--------------------|----------|
| Carrier frequency  | 2000 MHz |
| Antenna gain (eNB) | 15 dB    |
| MCL                | 80 dB    |

**Table 3.9: Simulation/transmission parameters specific for rural areas.**

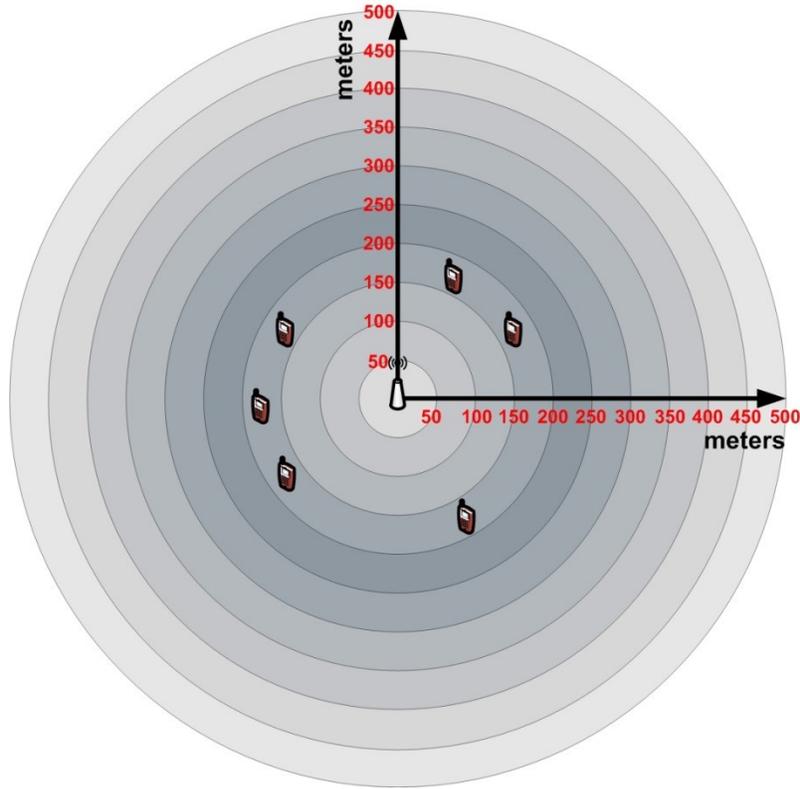
### 3.4.3 Simulation scenarios and results

To assess the performance of SSU, we conducted a series of system-level simulations. The algorithm aims to improve the data rates and overall throughput for cell-edge users;

therefore, the effectiveness of the algorithm could be evaluated as a function of the distance from the cell center. In each simulation run, the UEs are randomly placed within a narrow predefined range of distances measured from the center of the serving eNB cell. For instance, the first set of simulations was performed in an urban area of 2800 meters by 3000 meters, with 16 eNBs providing full geographical coverage. Figure 3.12 illustrates this setup. The grey circles around the eNBs are the separation distances (of 500m) and the red circles show the coverage areas of the eNBs (750m). The BS-UE distances were increased in each simulation run, in increments of 50 meters. In other words, the first simulation run positioned all the UEs within the first 50 meters around their serving eNBs, while the UEs in the second simulation run were placed between 50 and 100 meters from their serving eNBs, and so on. For example, in the fourth simulation run, all the UEs are placed between 150 and 200 meters away from their serving eNBs. This example is illustrated in Figure 3.13.



**Figure 3.12: Simulation area showing eNB positions.**

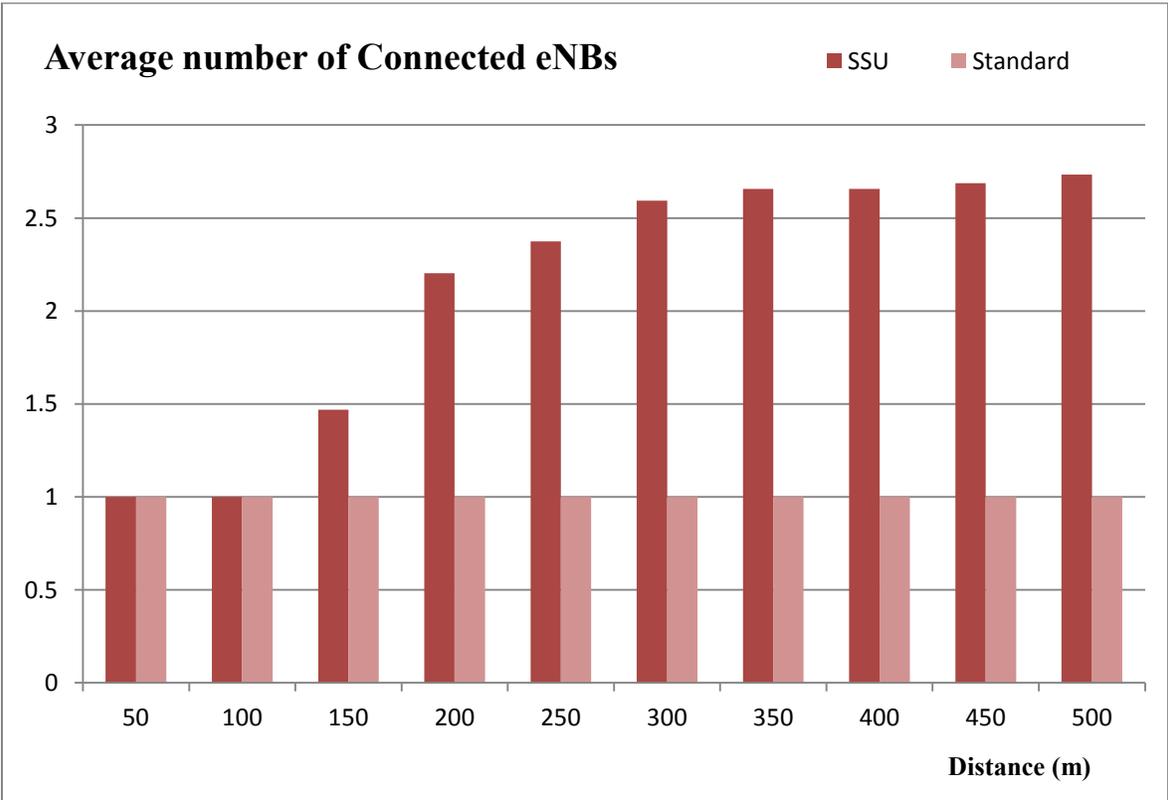


**Figure 3.13: Simulation scenario to evaluate SSU as a function of distance.**

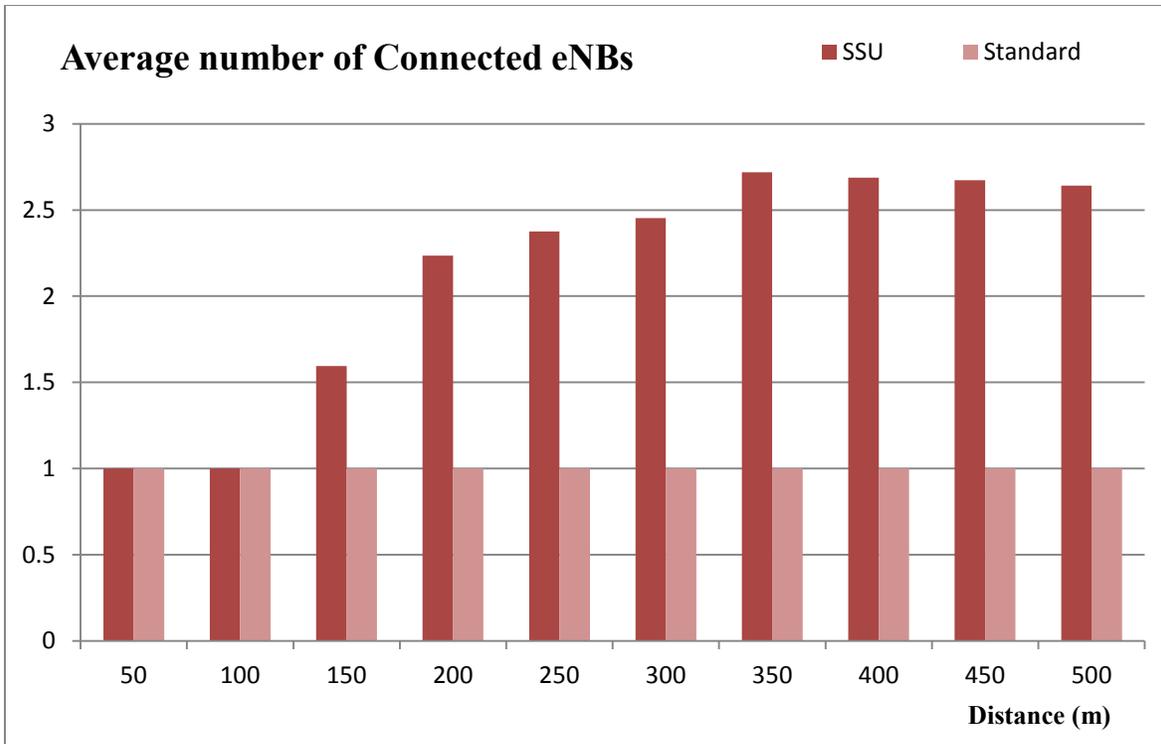
A user can only communicate with its serving BS when it travels through a cellular network that uses a conventional non-cooperative algorithm. This case is true even when the user is near the edges of a cell. On the other hand, SSU provides better performance by allowing the UE to communicate with more than one eNB when the user is close to the cell edges.

In Figure 3.14 to Figure 3.19, the horizontal axis represents the UEs' average distance from their serving eNBs. Figure 3.14 and Figure 3.15 show the average number of eNBs that each UE communicates with during the upload process, in an urban area setting with operating frequencies of 900 MHz and 2000 MHz, respectively. The UEs in the standard (using the conventional non-cooperative algorithm) simulation only connect to one eNB, regardless of their positions in the cell. The UEs in the SSU simulation behave in a similar manner when they are within a reasonable distance from their serving eNB. However, as these UEs move towards the edges of their cells, there is a higher probability for the UEs to communicate with neighboring eNB as well. When the UEs are within 50 meters from their cells' edges, they communicate

with an average of 2.73 eNBs in areas operating at 900 MHz and 2.64 eNBs in areas operating at 2000 MHz. Communicating with multiple eNBs has a number of effects on the network. First, there is a higher potential for collaboration between these eNBs using CoMP. This collaboration can be used to improve the upload performance for the UEs. However, this collaboration using SSU implies that eNBs communicate with each other and exchange both control and data messages over the X2 backhaul. This imposes an overhead on the network, especially for networks with low-bandwidth backhaul connections.

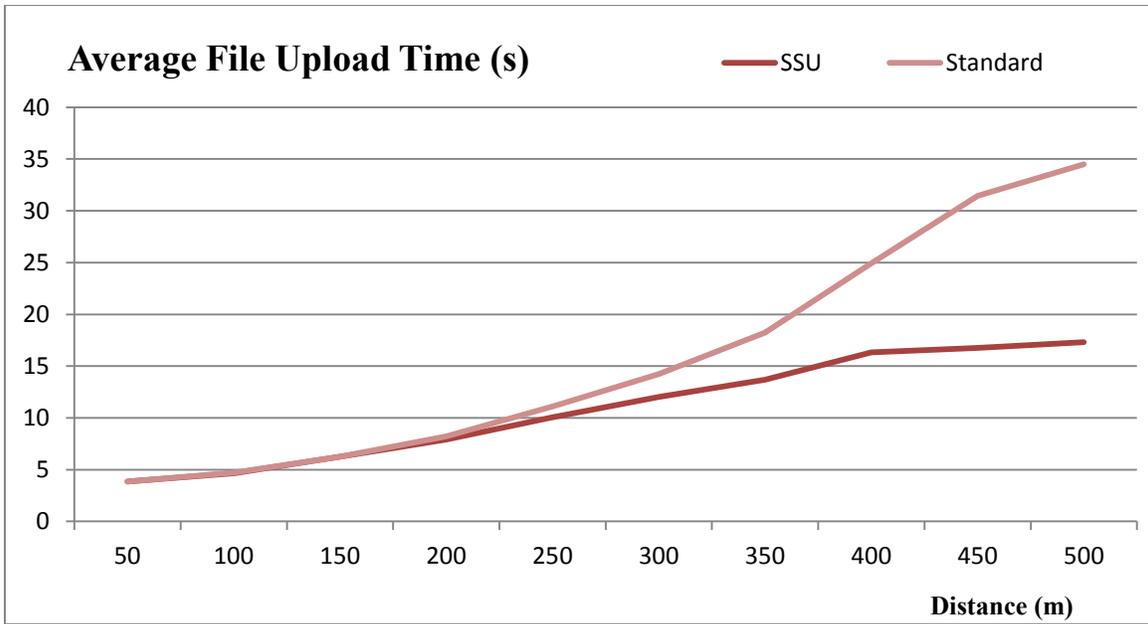


**Figure 3.14: Number of connected eNBs vs. distance from eNB (urban area, 900 MHz).**

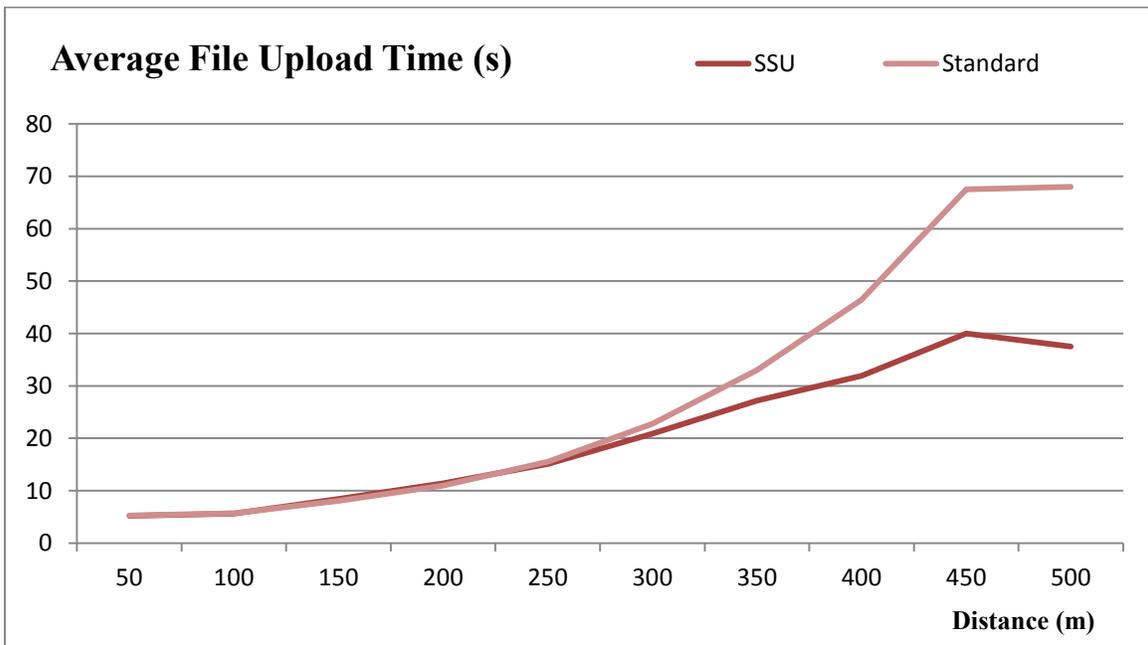


**Figure 3.15: Number of connected eNBs vs. distance from eNB (urban area, 2000 MHz).**

Figure 3.16 and Figure 3.17 show the average upload time versus the average UE distance from the center of their cells, for urban areas with operating frequencies of 900 MHz and 2000 MHz respectively. The upload time is the total time it takes to upload the file, which starts from the moment the UE creates and sends an *UploadRequest* message to the moment it receives a *Bitfield* message from its serving eNB. It can be seen from Figure 3.16 and Figure 3.17 that the time it takes for a UE employing the standard non-cooperative algorithm increases as the UE moves towards the cell edges. This occurs because when the UE is close to the cell edges, it receives a weaker signal from its serving eNB, as well as a higher level of interference from neighboring eNBs. Both these reasons contribute to the increase in the average upload time. UEs employing SSU experience the same trend as they approach the cell edges. However, the degradation in their upload performance is less severe, compared to UEs employing the standard non-cooperative algorithm. From Figure 3.16 and Figure 3.17, it can be seen that SSU becomes effective at improving the upload performance of UEs when the UEs are about 250 meters or further from the cell center. Closer to the center of the cell, SSU slightly increases the overhead, caused by the additional control messages.



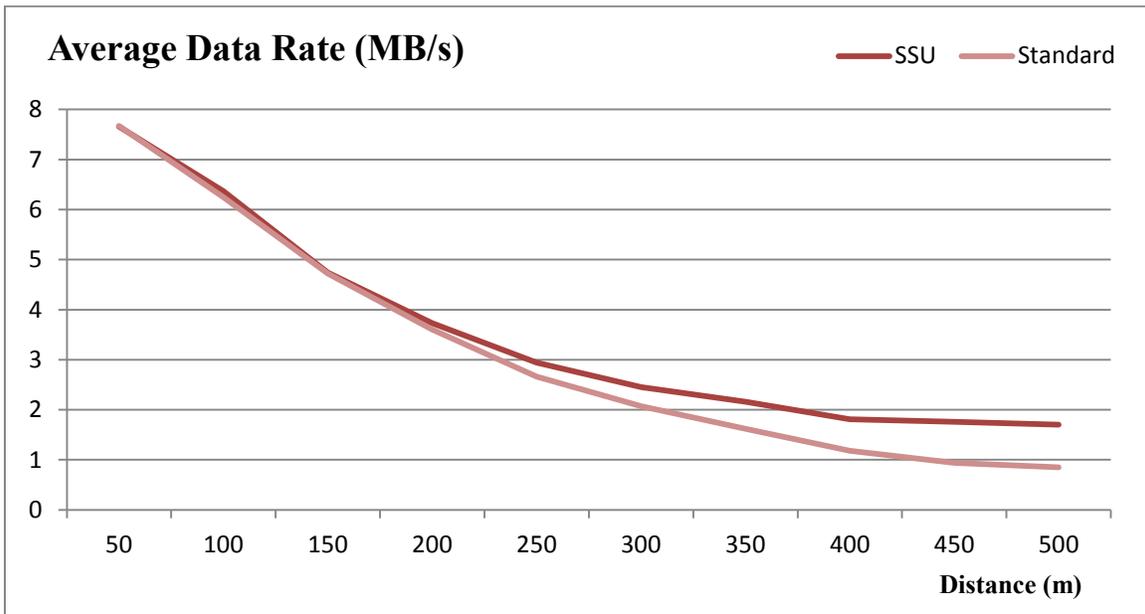
**Figure 3.16: File upload time vs. distance (urban areas, 900 MHz).**



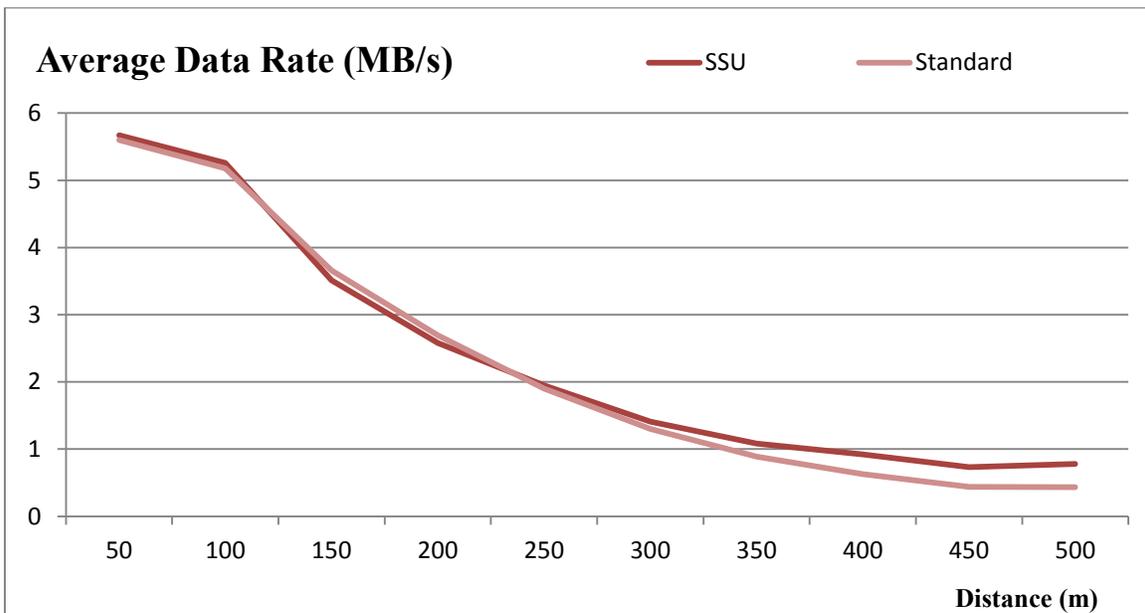
**Figure 3.17: File upload time vs. distance (urban areas, 2000 MHz).**

Figure 3.18 and Figure 3.19 show a comparison between SSU and the non-cooperative algorithm with respect to the average data rate they provide to the UEs as a function of their distance from the cell center, in urban areas with operating frequencies of 900 MHz and 2000 MHz, respectively. These figures show similar results to those shown in Figure 3.16 and Figure

3.17. According to these figures, both algorithms perform similarly when the UE is close to the center of the cell. As a UE moves towards the cell edges, the effects of SSU become more apparent on the average data rate. At the cell edges, SSU provides nearly twice the data rate compared to the conventional non-cooperative algorithm.



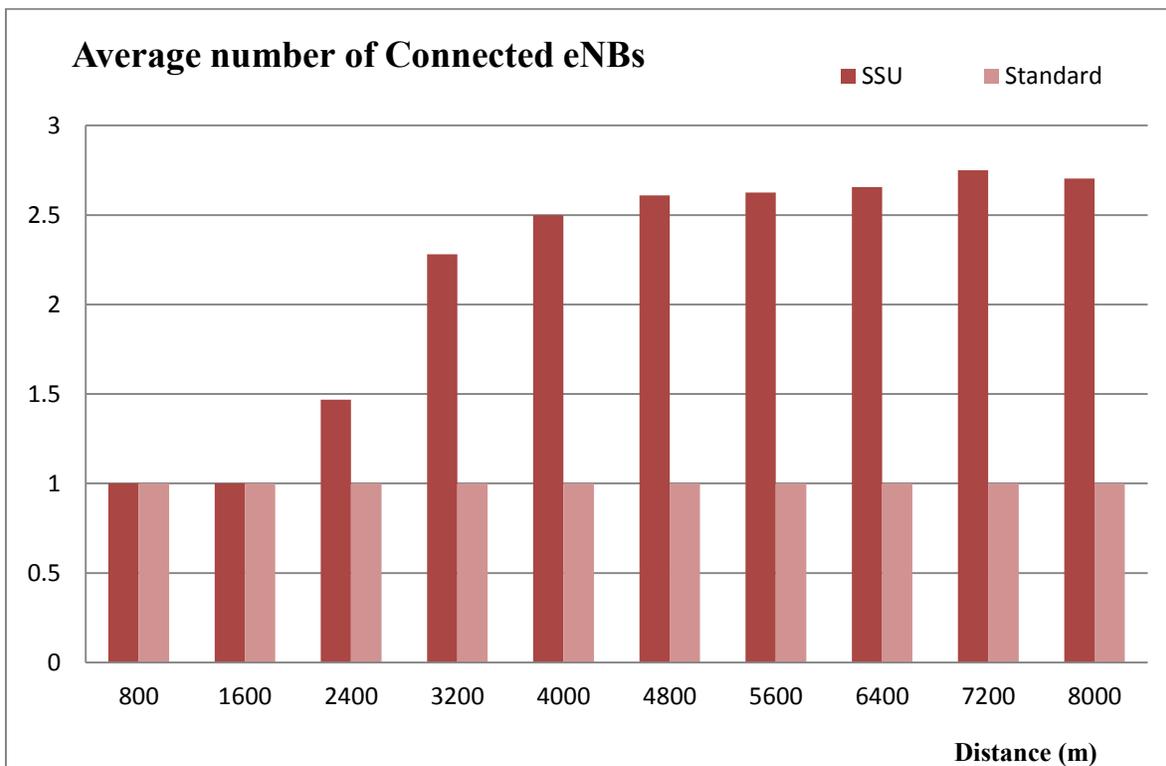
**Figure 3.18: Data rate vs. Distance from eNB (urban area, 900 MHz).**



**Figure 3.19: Data rate vs. Distance from eNB (urban area, 2000 MHz).**

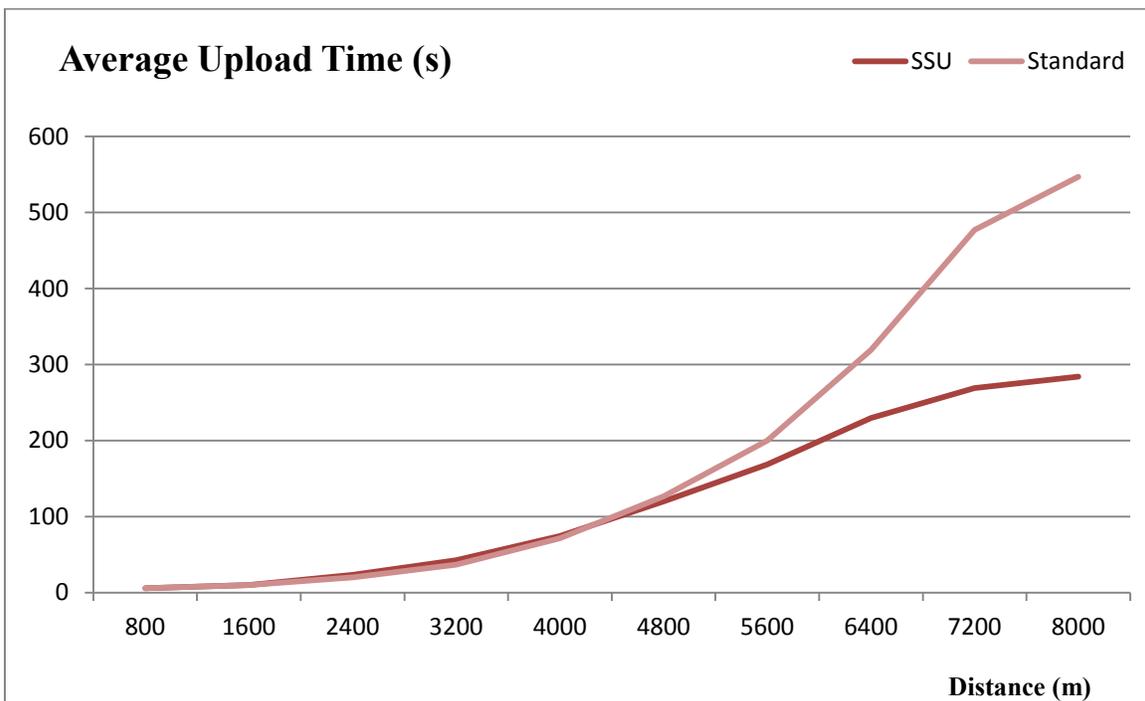
A similar scenario was used for rural areas where the operating area was set to 8 kilometers by 8 kilometers, and the BS-UE separation distances were increased in increments of 800 meters. In other words, the UEs were positioned in the first 800 meters around the eNBs in the first simulation run, and between 800 meters and 1.6 kilometers in the second simulation run, and so on. There were 64 active UEs in the network in each simulation run, and each UE was triggered to begin the upload of one data file at the beginning of the simulation. The simulations were allowed to run until all the file uploads were complete and the data was collected.

Figure 3.20 shows the average number of eNBs that the UEs communicate with during the simulation as a function of their distance from their serving eNB. The figure shows that UEs employing the standard non-cooperative algorithm only communicate with their serving eNB, regardless of their position within the cell. On the other hand, in a network employing SSU, the number of eNBs that a UE communicates with increases as the UE moves closer to the cell edges. At the edge of the cell, a UE using SSU is able to communicate with 2.7 eNBs on average.



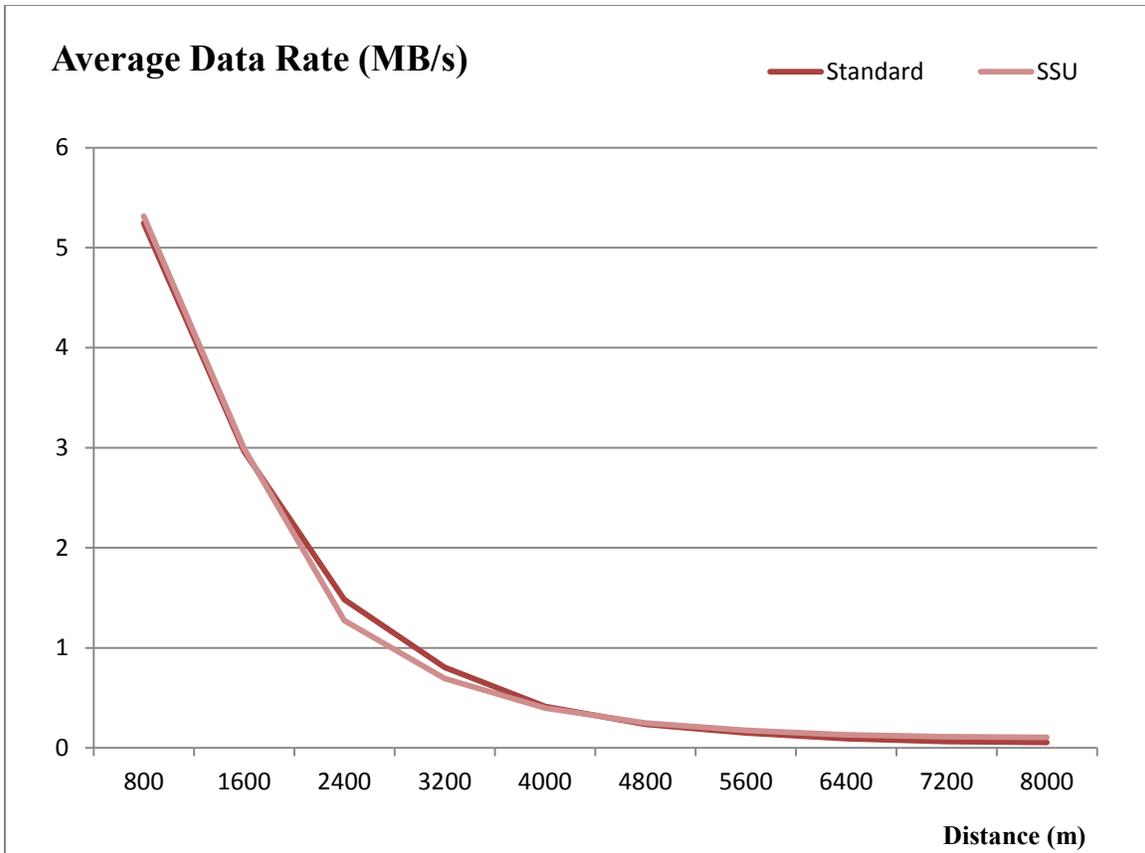
**Figure 3.20: Number of connected eNBs vs. distance from eNB (rural area).**

Figure 3.21 shows the average file upload time (in seconds) of SSU and the conventional non-cooperative algorithm versus the distance of the UEs from their serving eNBs. Similarly, Figure 3.22 shows a comparison between the two algorithms with respect to the data rate they provide to the UEs. According to Figure 3.21, SSU provides significant reduction in the average file upload times. Near the edges of the cell, SSU was able to reduce the average upload time to approximately half the time required by UEs employing the conventional non-cooperative algorithm. The effect of SSU becomes apparent when the UEs are around 4400 meters away from their serving eNBs.



**Figure 3.21: Average file upload time vs. distance from eNB (rural area).**

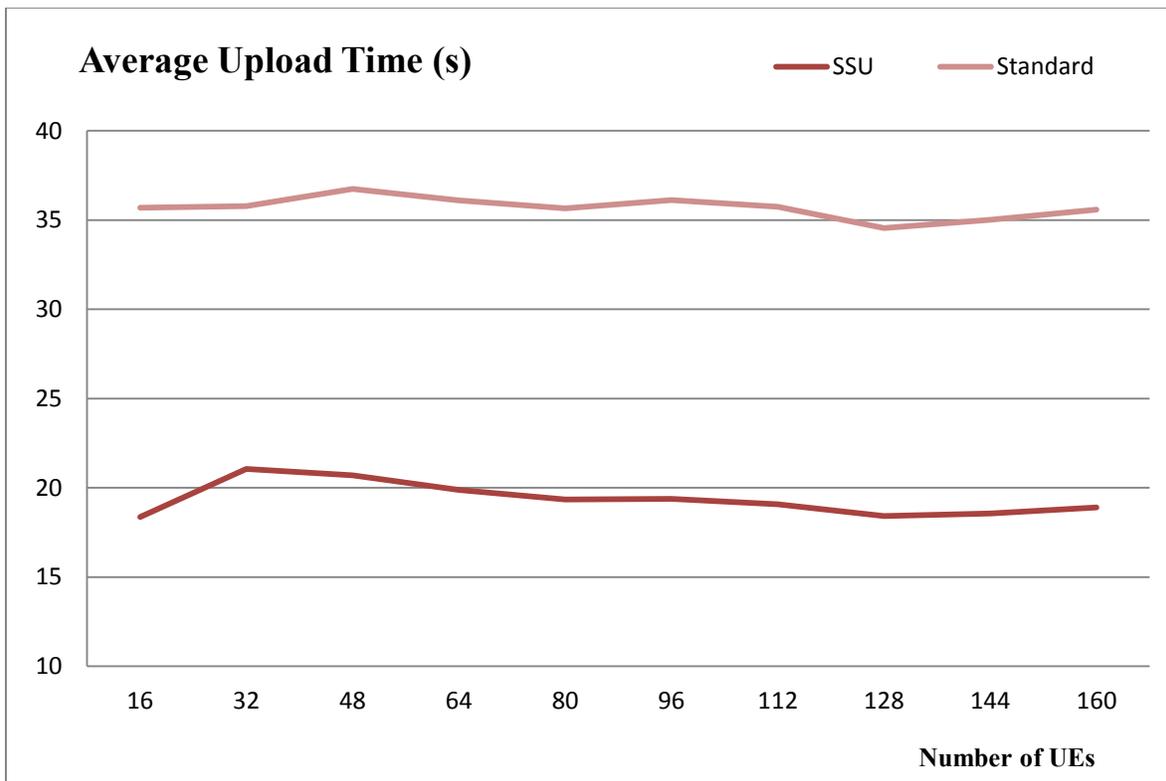
Figure 3.22 shows similar results with respect to the average data rate. At the cell edges, SSU provides 92% increase in the data rate available for UEs. However, to achieve this improvement, SSU applies some level of signalling overhead on the network. This overhead is visible when the UEs are closer to the cell center, where the performance of SSU is lower than that of the conventional non-cooperative algorithm.



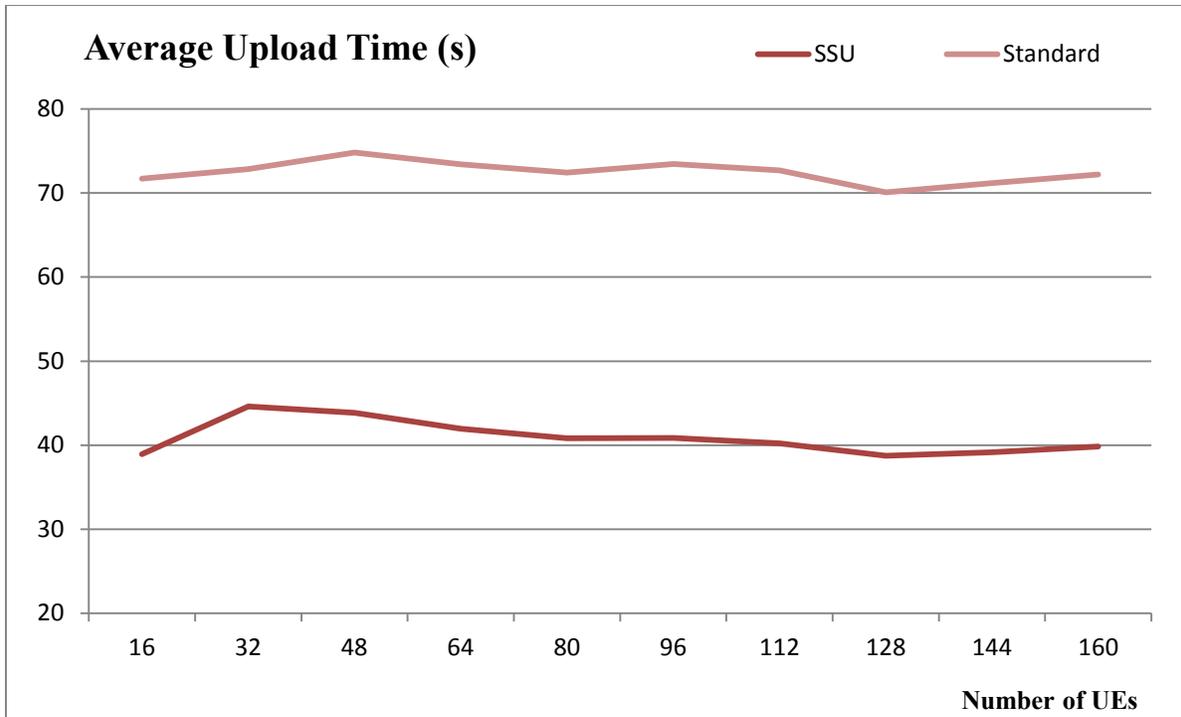
**Figure 3.22: Average data rate vs. distance from eNB (rural area).**

A second set of simulations were conducted to study the effect of increasing the number of cell-edge UEs on the performance of SSU. We considered a limited scale scenario in an urban area of 2800 meters by 3000 meters. The area required 16 eNBs to provide full geographical coverage. We began with 16 cell-edge UEs in the first round of simulations and we increased the number of the UEs in each of the following iterations in increments of 16. In the final round of simulations, there were 160 active cell-edge UEs in the simulated area. In a CoMP scenario, as the number of cell-edge UEs increases, the number of required coordination sets increases proportionally. This means that an eNB may need to deal with more UEs (the UEs served by this eNB, as well as UEs from neighboring cells), compared to a conventional non-cooperative algorithm in which an eNB is only required to serve UEs within the cell). The increase in the number of coordination sets imposes an overhead on the eNBs and the X2 backhaul links. The goal of this set of simulations is to study how an increase in the number of cell-edge UEs affects the performance of SSU.

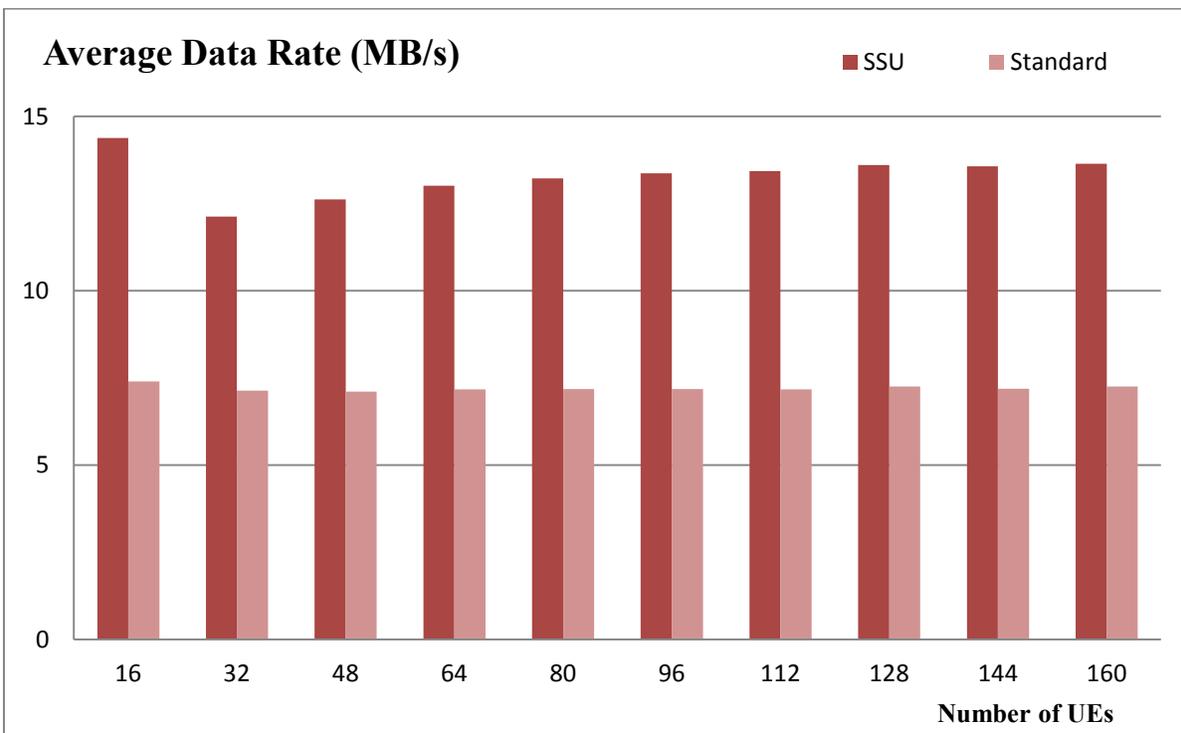
In Figure 3.23 to Figure 3.26, the number of UEs in the simulated area is shown on the horizontal axis. Figure 3.23 and Figure 3.24 show the average file upload time versus the total number of cell-edge UEs in the coverage area. The figures show that SSU provides an overall improved upload performance compared to the conventional non-cooperative scheme. Moreover, the overhead imposed by SSU does lead to a degradation of its performance as the number of UEs increases. The variations in the upload time are caused by the variations in the initial positions of the UEs. Figures 3.25 and 3.26 provide similar conclusions regarding the performance of SSU. These figures show that SSU provides lower latency and higher data rates for its users, compared to the standard method. Moreover, the figures show that SSU is able to offer a consistent quality of service as the number of UEs in the network increases.



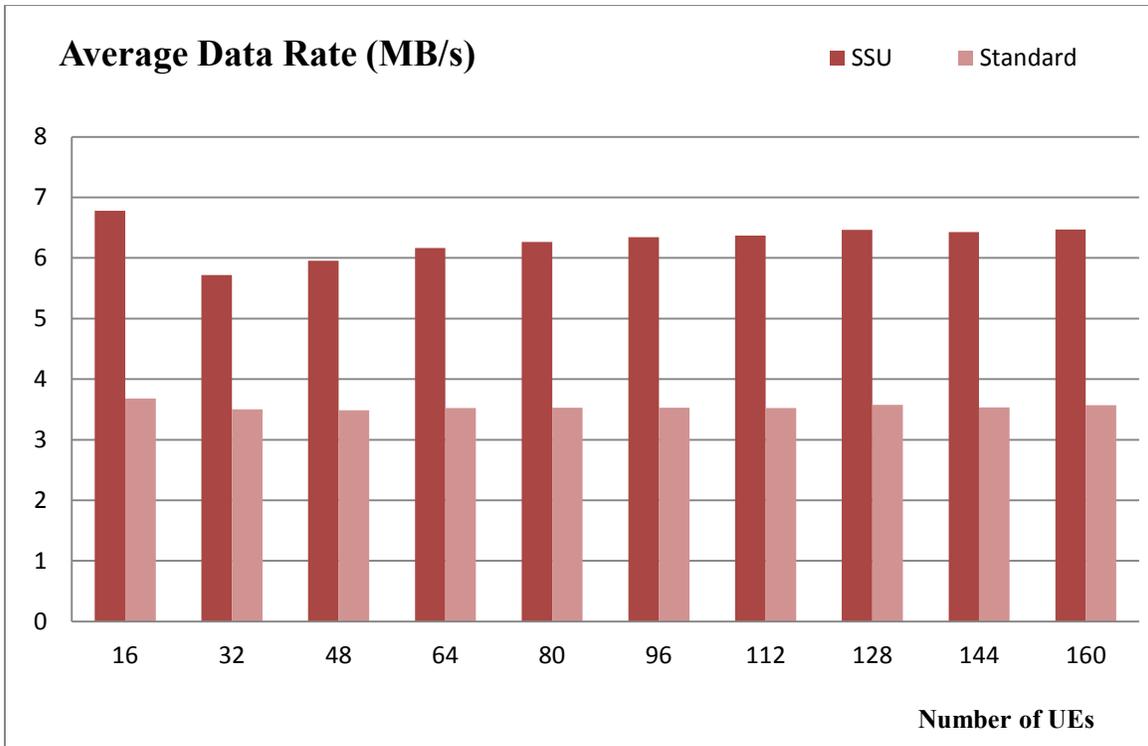
**Figure 3.23: Upload time vs. number of UEs (urban area, 900 MHz).**



**Figure 3.24: Upload time vs. number of UEs (urban area, 2000 MHz).**



**Figure 3.25: Data rate vs. number of UEs (urban area, 900 MHz).**



**Figure 3.26: Data rate vs. number of UEs (urban area, 2000 MHz).**

### 3.5 Summary

In this chapter, we introduced Shared Segmented Upload (SSU) as a novel algorithm for LTE-A networks that rely on Coordinated MultiPoint techniques to improve the uplink performance for UEs located near the cell edge. Moreover, we presented a simplified DEVS model of an LTE network employing SSU as well as a conventional non-cooperative algorithm. The simulation results show that SSU is able to provide significant improvement in the upload performance of UEs, especially as they move closer to the edges of their cells.

## Chapter 4

# Upload User Collaboration

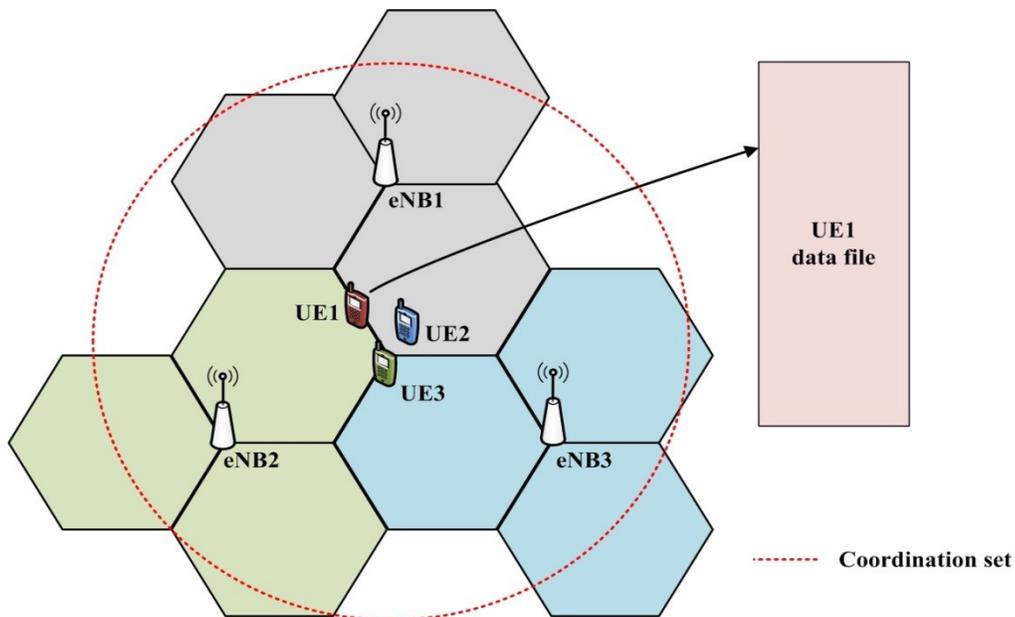
The continually increasing demand for wireless communication networks to provide higher data rates and improved spectral efficiency has led to the development of a number of standards and technologies. As previously mentioned, LTE introduced a number of technologies to meet the requirements in IMT-2000. These technologies include MIMO, Enhanced ICIC, carrier aggregation, heterogeneous deployments, etc. More recently, Device-to-Device (D2D) communication was introduced in the LTE-A standard in an effort to provide higher throughput, lower energy consumption, and a better quality of service for users in close proximity who wish to communicate with each other. D2D allows UEs to directly communicate with each other using licensed frequency bands in order to minimize interference [66].

In this chapter, we present another algorithm that aims to improve the upload performance of UEs in a distributed CoMP scenario. User Upload Collaboration (UCC) shares a few similarities with SSU where a UE is able to upload a data file in a series of pieces to multiple eNBs participating in a CoMP coordination set. However, UCC also utilizes LTE-A's D2D capabilities to allow a UE to request assistance with the file upload from nearby UEs. The collaboration between the UEs improves the overall upload performance by combining the upload powers of multiple UEs. This method can also be applied to downlink transmissions, as well as non-CoMP scenarios as long as UEs are located in close proximity of each other.

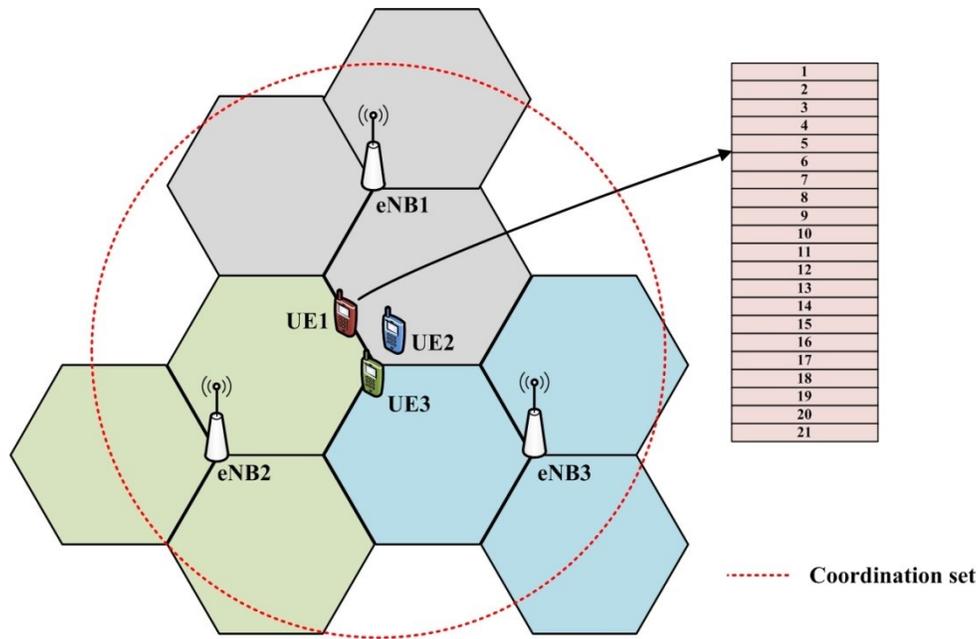
## 4.1 Algorithm Overview

UUC focuses on enhancing a UE's upload process by using the upload power of multiple users that are sufficiently close to each other. The UE that wishes to upload a file (we will call that UE "owner UE" from now on) splits the data file into a series of pieces and uploads them to the eNBs participating in the UE's CoMP coordination set. Parallel to the UE-to-eNB upload, the owner UE also requests assistance from neighboring UEs. A neighboring UE that is willing to assist with the file upload (from now on called the "helper UE") can then receive a portion of the original data file from the owner UE directly over a D2D connection. The helper UE would then upload the received pieces to the eNBs in its CoMP coordination set as if the pieces were part of its own file.

To explain the upload process in more detail, consider a network of three eNBs and three UEs. The scenario is illustrated in Figure 4.1. One of the UEs (UE1) located at the cell edge wishes to upload a large data file to the network. The three eNBs in its range form a CoMP coordination set, with eNB1 acting as UE1's serving eNB. In addition, two other UEs (UE2 and UE3) are within close proximity to UE1 and are able to communicate directly with UE1 over D2D channels. Before the upload begins, UE1 splits the data file into a number of pieces (see Figure 4.2).



**Figure 4.1: UE1 wishes to upload a large data file.**



**Figure 4.2: UE1 splits the data file into a series of pieces.**

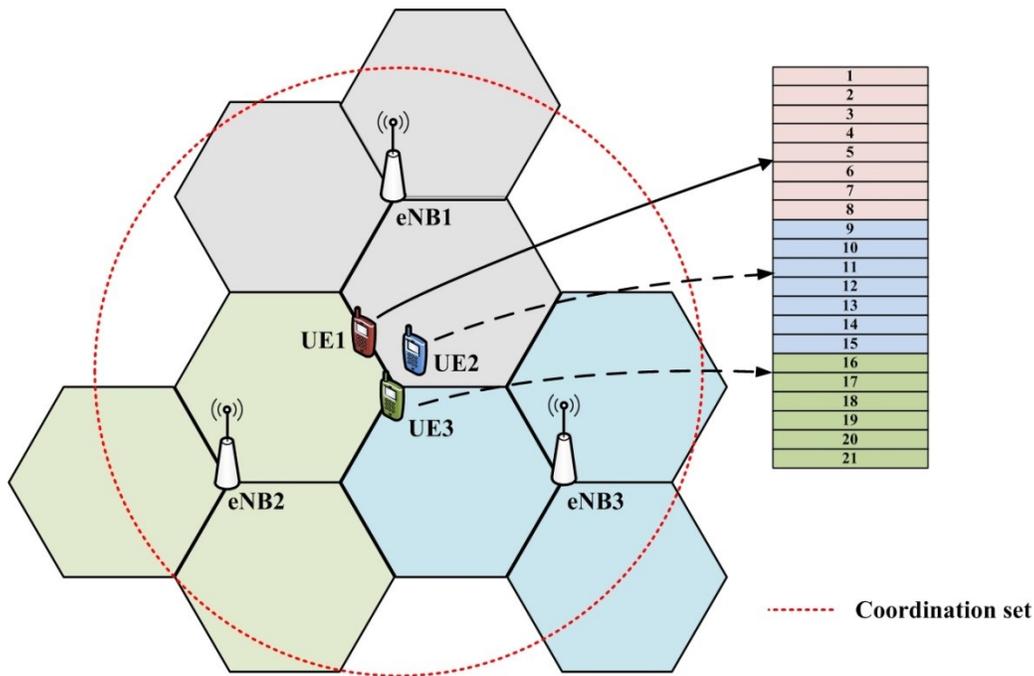
The upload begins in a similar manner as Shared Segmented Upload with some modifications. First, the owner UE (UE1) sends an *UploadRequest* message to the eNBs in its range. The eNBs that are willing to participate in the coordination set send a *Handshake* message to the owner UE. This decision usually depends on the quality of the transmission link between the owner UE and the eNB, as well as the traffic load on the eNB.

As soon as the owner UE receives a Handshake message from its serving eNB, the owner UE creates a *Data File Descriptor* (or *DFD*) message and sends it to its serving eNB. The *DFD* message is similar to the *MetaInfo* file used in SSU. It contains details about the owner UE, its position, as well as details about the data file the UE wishes to upload. The *DFD* message indicates how many pieces need to be uploaded, and specifies the start and end bytes of each of the pieces. This allows the file to be split into pieces of different sizes, depending on the quality of the uplink and the D2D transmission links. Compared to the size of the original data file, the resulting *DFD* message is relatively small, and can be easily sent to the serving eNB. The owner UE sends the same *DFD* message to other non-serving eNBs that join the coordination set.

The owner UE begins uploading pieces to the eNBs in the coordination set once it receives an acknowledgement from serving eNB indicating the receipt of the *DFD* message. The

pieces are sent in a series of Transport Blocks (TBs) to the eNBs in the coordination set, simultaneously, using Joint Transmission techniques. The TBs are sent using a “multicast” approach so that the eNBs in the coordination set receive the same copy of the TB at the same time.

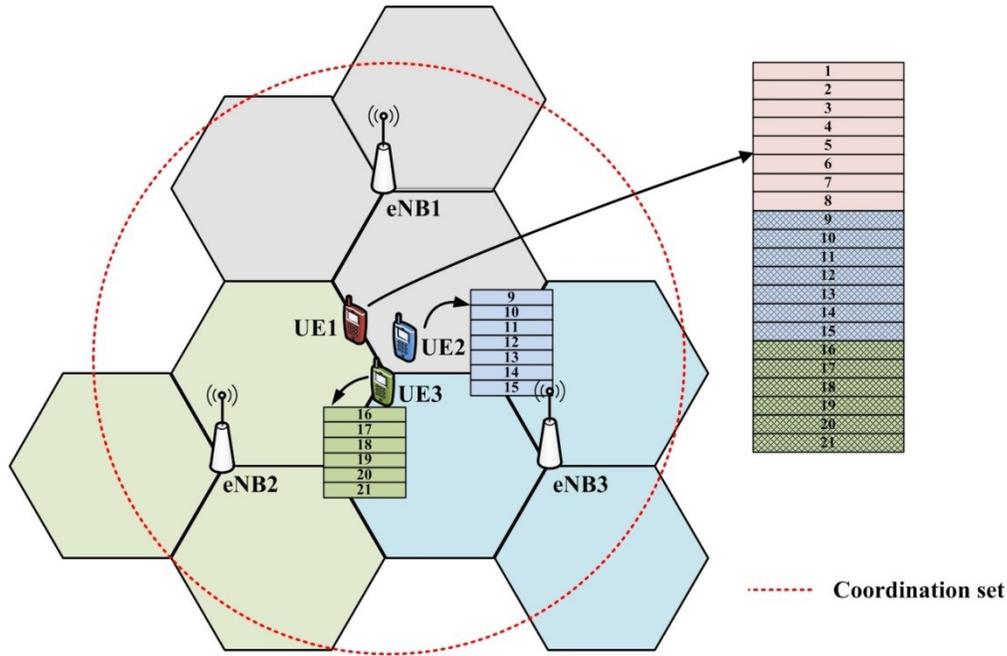
Similarly, at the beginning of the upload process, UE1 also sends *UploadAssistance* messages to the neighboring UEs (UE2 and UE3) to request assistance with the file upload. Neighboring UEs that are willing to assist with the upload process, request resources from their own coordination set of eNBs using the same process as the owner UE. In other words, each helper UE sends an *UploadRequest* message to the eNBs in range to begin the upload process, as if the data was their own. Once a helper UE receives a *Handshake* message from its serving eNB, it sends a *Confirmation* message to the owner UE. When UE1 receives a *Confirmation* messages, it assigns a set of pieces to that helper UE (see Figure 4.3) and begins forwarding these pieces to the helper UE over D2D channels, in parallel to the regular upload.



**Figure 4.3: UE1 assigns a set of pieces to each of the helper UEs.**

The owner UE continues transmitting the assigned pieces to each of the helper UEs in a round robin manner. As soon as the helper UE receives the first piece, it begins the upload of that

piece to its coordination set of eNBs over its own transmission channels, using SSU. This approach maximizes the number of helper UEs simultaneously uploading data to the network and reduces the overall time required to upload the file. The owner UE is no longer required to upload the entire file (Figure 4.4).



**Figure 4.4: UE1 is no longer required to upload the entire file.**

Pieces of the data file are uploaded in TBs, either by the owner UE itself or with the help of one or more helper UEs. The TBs are received by the serving eNB of the owner UE, as well as other non-serving eNBs. However, the entire upload process is managed by the owner UE's serving eNB (eNB1 in the example scenario). The non-serving eNBs that receive a TB, with or without errors, send a *TBStatus* message to the serving eNB (eNB1). The *TBStatus* message informs the serving eNB which of the eNBs received the error-free TB. The serving eNB uses this information to keep track of the receipt status of the TBs, and to avoid unnecessary retransmission, it requests if another eNB has successfully received the TB. The serving eNB sends an acknowledgement message (ACK/NACK) to the owner UE once it receives all the TBs (or *TBStatus* messages) of a piece. Moreover, if the received piece was uploaded through a helper UE, the owner UE's serving eNB sends an acknowledgement message to that helper UE as well.

When the last TB of a piece has been received, the serving eNB uses the receipt status information to select one of the eNBs to send the full piece to the MME, or another entity at the same level. The selected eNB is called the transporter eNB and is responsible for gathering the TBs from non-serving eNBs and forwarding the piece to the MME. This is only required in the case that none of the eNBs has received all the TBs of the piece. The selection of the transporter eNB depends on the number of TBs that have been received without errors at each of the participating eNBs. The serving eNB informs the selected transporter eNB by sending a *ForwardToMME* message with a list of missing TBs that the transporter eNB did not receive. Moreover, the serving eNB sends *ForwardMissingTBs* messages to other eNBs that have successfully received the missing TBs. These eNBs then forward the missing TBs to the transporter eNB over the X2 backhaul interface. The transporter eNB collects the missing TBs (if any), reconstructs the file piece, and forwards the entire piece to the MME over the S1 backhaul link.

At the end of the upload process, the file pieces (those uploaded by the owner UE as well as those uploaded by the helper UEs) are gathered at the Mobility Management Entity (MME), or another entity at the same level. The gathered pieces are then used to reconstruct the original data file.

Uploading a data file using this approach allows different pieces of the file to be uploaded through multiple transmission links. The approach aims to speed up the upload process of a data file regardless of the position of the owner UE. Therefore, UUC can be used to improve the upload performance of both cell-edge UEs and non-cell-edge UEs. The upload process using UUC is summarized in Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8. The figures illustrate the upload process using sequence diagrams. Figure 4.5 shows the initial steps of the upload process, which includes requesting resources from eNBs as well as assistance from neighboring UEs. Figure 4.6 shows how the owner UE uploads pieces to the eNBs in the coordination set while Figure 4.7 illustrates the role of the helper UEs in the upload process. Figure 4.8 shows the process used to gather TBs and file pieces to reconstruct the original data file at the eNBs.

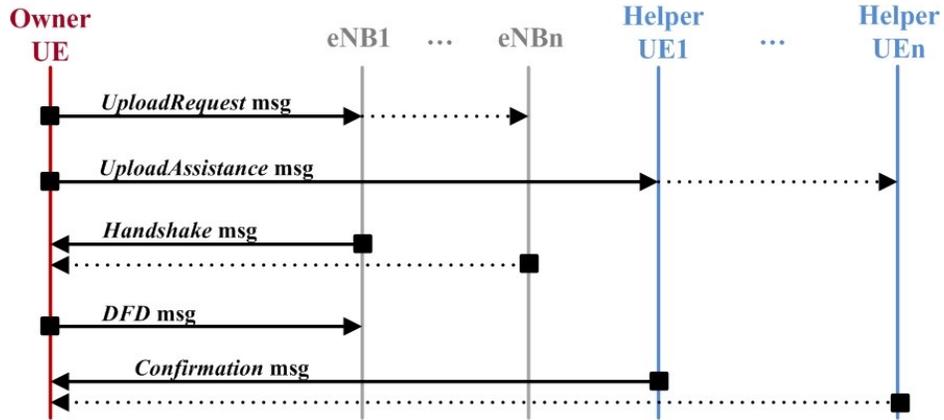


Figure 4.5: Initial steps of UUC (by the owner UE).

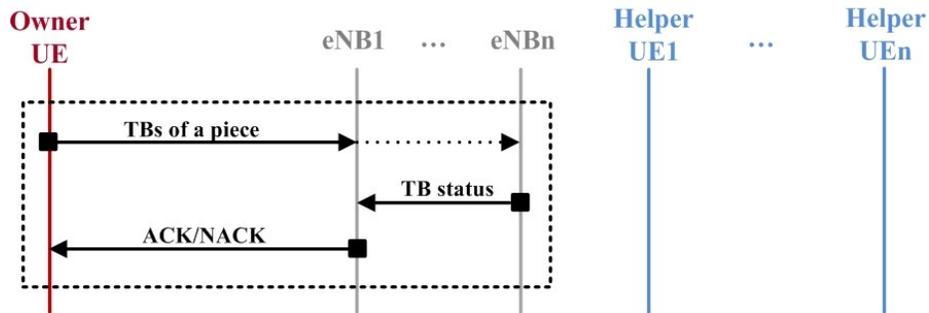


Figure 4.6: Upload of pieces by the owner UE.

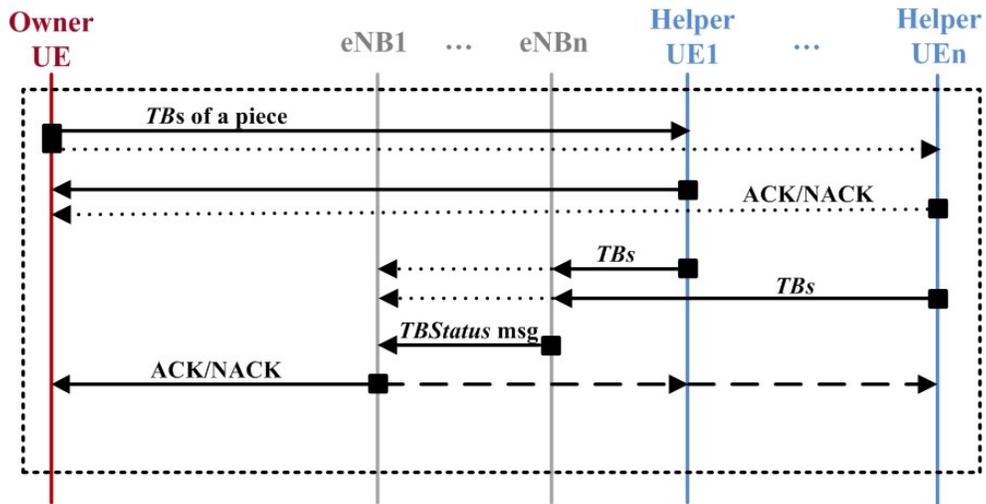
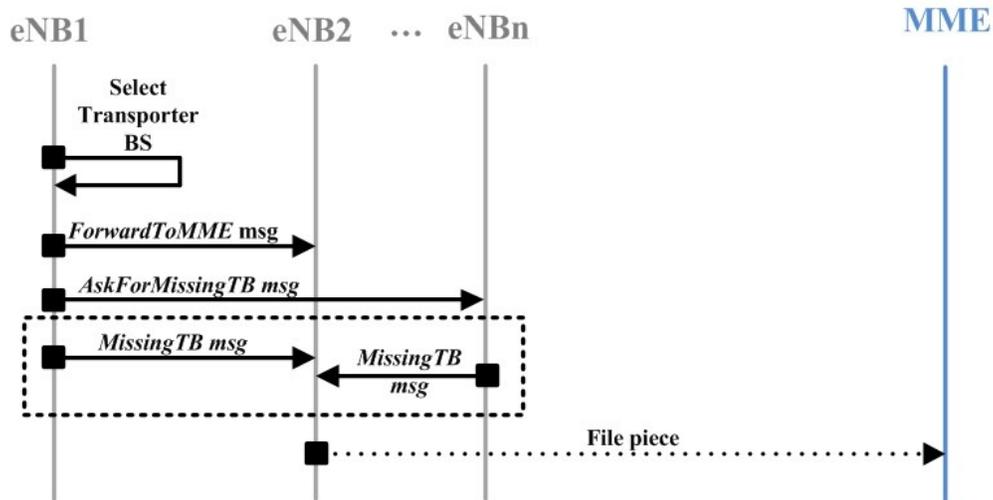


Figure 4.7: Role of the helper UEs in the upload process.



**Figure 4.8: Process of gathering TBs and pieces.**

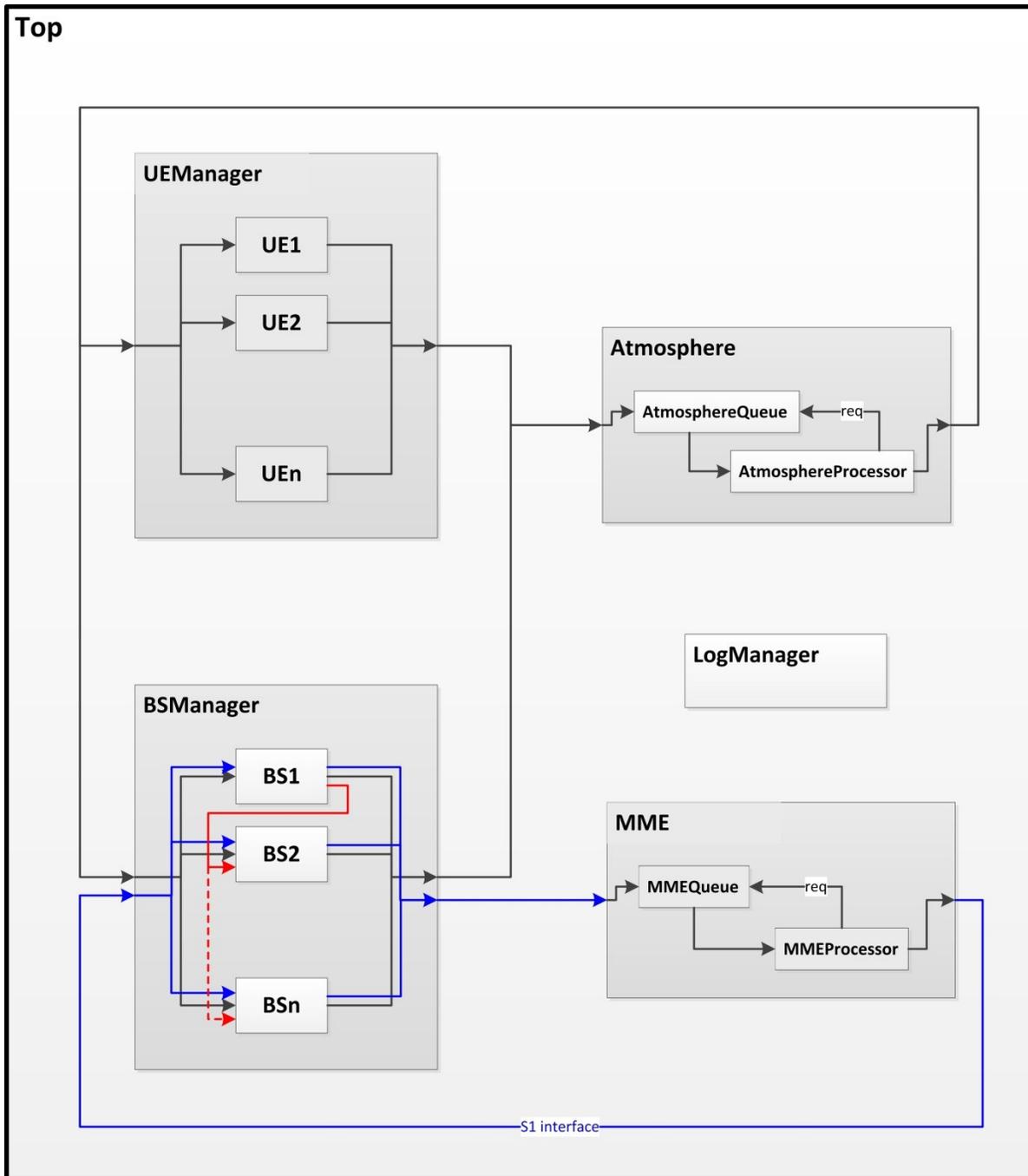
With respect to the size of the pieces, three different cases are considered: Uniform Piece Size (UPS), Variable Piece Size (VPS), and Dynamic Piece Size (DPS). In the first case, the file is split into equal-sized pieces (except for the last piece, which is always smaller than or equal to the remaining pieces). There is a trade-off between the size of the pieces and the efficiency of the algorithm. A small piece size leads to a large number of pieces. This imposes a signaling overhead on the network, and reduces the efficiency of the algorithm. On the other hand, a large piece size would minimize the achievable gains of the algorithm, as it would be similar to uploading a large file. In this thesis, we assume a uniform piece size of 1024 KB. In VPS, the owner UE considers various piece sizes and splits the file into pieces based on these sizes. During the upload process, the owner UE can choose one of the available piece sizes based on the helper UE's preference, or based on other factors such as the D2D channel conditions. Finally, in DPS, the owner UE determines the piece size during the upload process based on the concurrent channel conditions. In this case, the piece size would be unknown at the beginning of the upload process and therefore, the *DFD* message would not contain information about the pieces. Instead, it would only contain information about the file. As the owner UE creates pieces during the upload process, it populates the *DFD* message with information about the pieces. At the end of the upload process, the owner UE sends the complete *DFD* message to the serving eNB. The serving eNB uses the final *DFD* message to verify the correct transmission of the pieces, and the MME uses it to reconstruct the original data file.

Once the serving eNB receives confirmation that all the pieces of the data file have been received successfully by the eNBs in the network, it sends an *UploadFinished* message to the owner UE informing it that the upload process has been completed.

## 4.2 Modeling of UUC using DEVS

A DEVS model for an LTE network was developed in order to study the performance of UUC. The model's top-level design is based on the model used to test SSU. However, this model was designed to be flexible, in such a way that the basic functions of LTE systems are separated from the behaviour of the algorithm being tested. For example, an eNB operating on an LTE network would have to manage "communication sessions" between multiple UEs, regardless of the algorithm under consideration. Moreover, the layers of the LTE protocol stack have been implemented as separate model entities, allowing the behaviour of an algorithm to be modelled in more detail at each layer. The design of the model will be explained in detail in this section, and this design is one of the contributions of this thesis.

The structure of the model is shown in Figure 4.9. The top-level coupled model is similar to model discussed in Chapter 3. The top-level model is called *Area*, and contains one atomic model (*LogManager*) and four other coupled models (*UEManager*, *BSManager*, *MME*, and *Atmosphere*). The compositions and behaviours of *LogManager*, *UEManager*, and *BSManager* are identical to those in the previous model. The *Atmosphere* coupled model is also used to simplify the interconnections between UEs and BSs. However, in this model, each BS coupled model is directly connected to neighboring BS models through individual links (illustrated in Figure 4.9 using red links). The *MME* coupled model, as the name suggests, models the simplified behaviour of an MME entity, and is composed of two atomic models: *MMEQueue* and *MMEProcessor*. The *MMEQueue* atomic model is used to queue incoming messages to the MME, while *MMEProcessor* is used to process incoming messages, one at a time.



**Figure 4.9: Top-level model structure.**

Each UE and BS coupled model in Figure 4.9 is composed of a number of atomic and coupled submodels. The structure of the UE coupled model is shown in Figure 4.10. Each of the four layers in the user plane of the LTE protocol stack (PDCP, RLC, MAC and PHY layers) are implemented as separate entities. Moreover, an application layer entity (*UEAPPLayer*) was

added to initiate the file upload process at the beginning of the simulations. Each layer is associated with an instance of a *UEQueue* to queue up the incoming messages. A *UETimer* atomic model is used to issue a “tick” signal at regular intervals. The signal is used by the *UEPHYLayer* atomic model to synchronize the processing of incoming messages with LTE’s Transmission Time Intervals (TTIs) of 1 millisecond. TTI refers to the smallest time period used for scheduling resources and transmitting data blocks. In LTE, a TTI of 1 millisecond is used [35]. The UE coupled model also contains two other coupled models, namely *UEFilter* and *UETransmitter*. *UEFilter* is used to filter messages received by the *UE* coupled model from the *Atmosphere* model through the *fromATM* input port. Since these messages are broadcast in nature, each receiver (UE or BS) would need to filter these messages based on the intended destination of each message. The *UEFilter* coupled model is composed of a *UEFilterQueue* and *UEFilterProcessor*. The *UETransmitter* coupled model is used to synchronize the outgoing messages sent by the UE with LTE’s TTI. In other words, the *UETransmitter*’s task is to queue the outgoing messages and send them at 1 millisecond intervals according to the scheduled resources available to the UE. Similar to the *UEFilter* coupled model, the *UETransmitter* coupled model is composed of two atomic models: *UETransmitterQueue* and *UETransmitterProcessor*. In both cases, the queue atomic model queues incoming messages for the processor, which is tasked with processing these messages one at a time.

*BS* coupled models have a similar structure (see Figure 4.11). In addition to the *fromATM* input port that receives all messages sent through the *Atmosphere* entity, *BS* coupled models also receive messages from the *MME* entity through the *fromMME* input port, as well as messages from its neighbouring BSs through the *fromNeighbourX* input ports, where  $X$  is the neighbour number ( $1 \leq X \leq n$ ). The model was designed to accommodate any number of neighbouring BSs, and therefore, allows it to be easily adapted to support heterogeneous network deployments. Similarly, *BS* coupled models have matching outputs ports to the *MME* model as well as each of its  $n$  neighboring BSs.

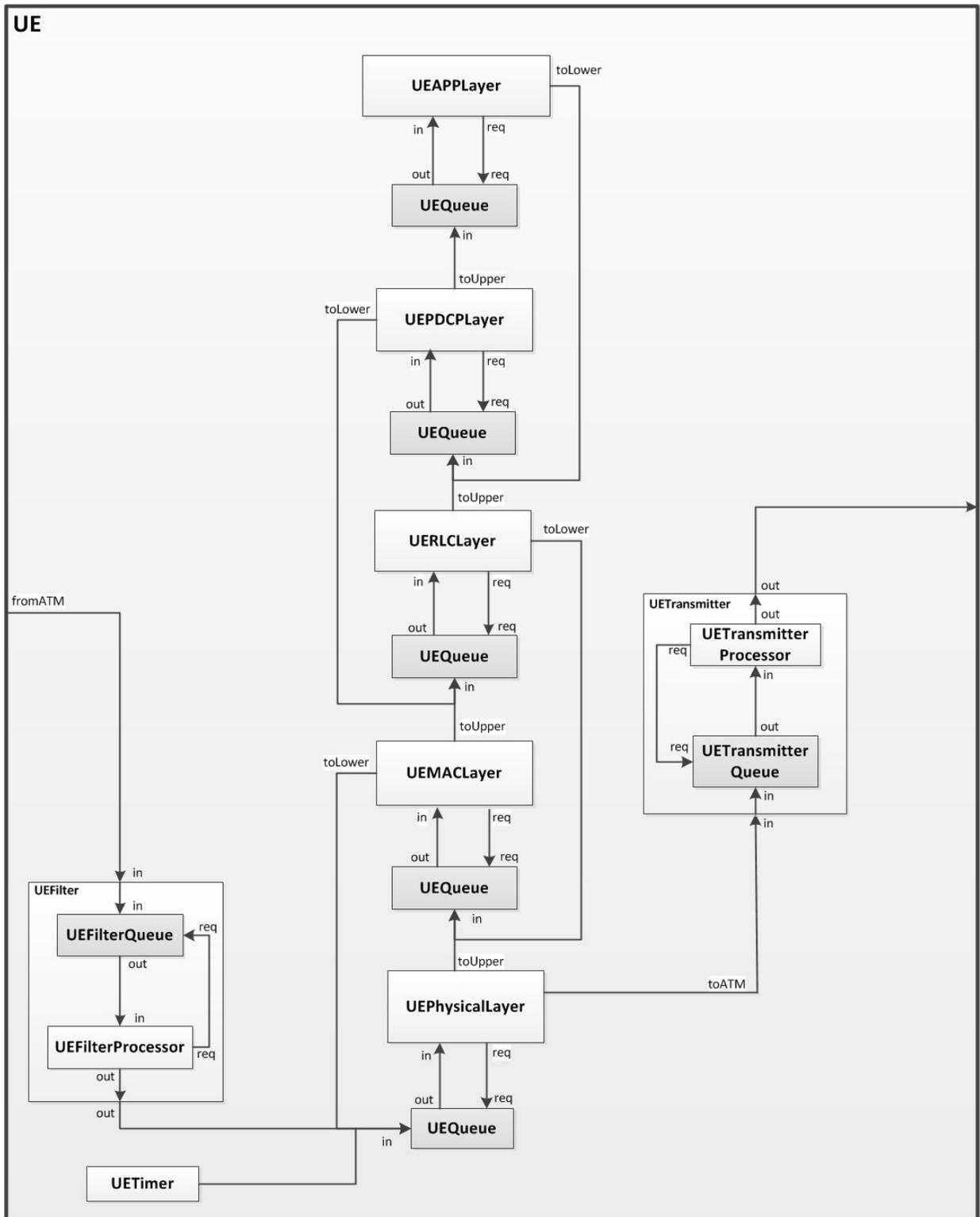


Figure 4.10: Structure of the UE coupled model showing atomic and coupled submodels.

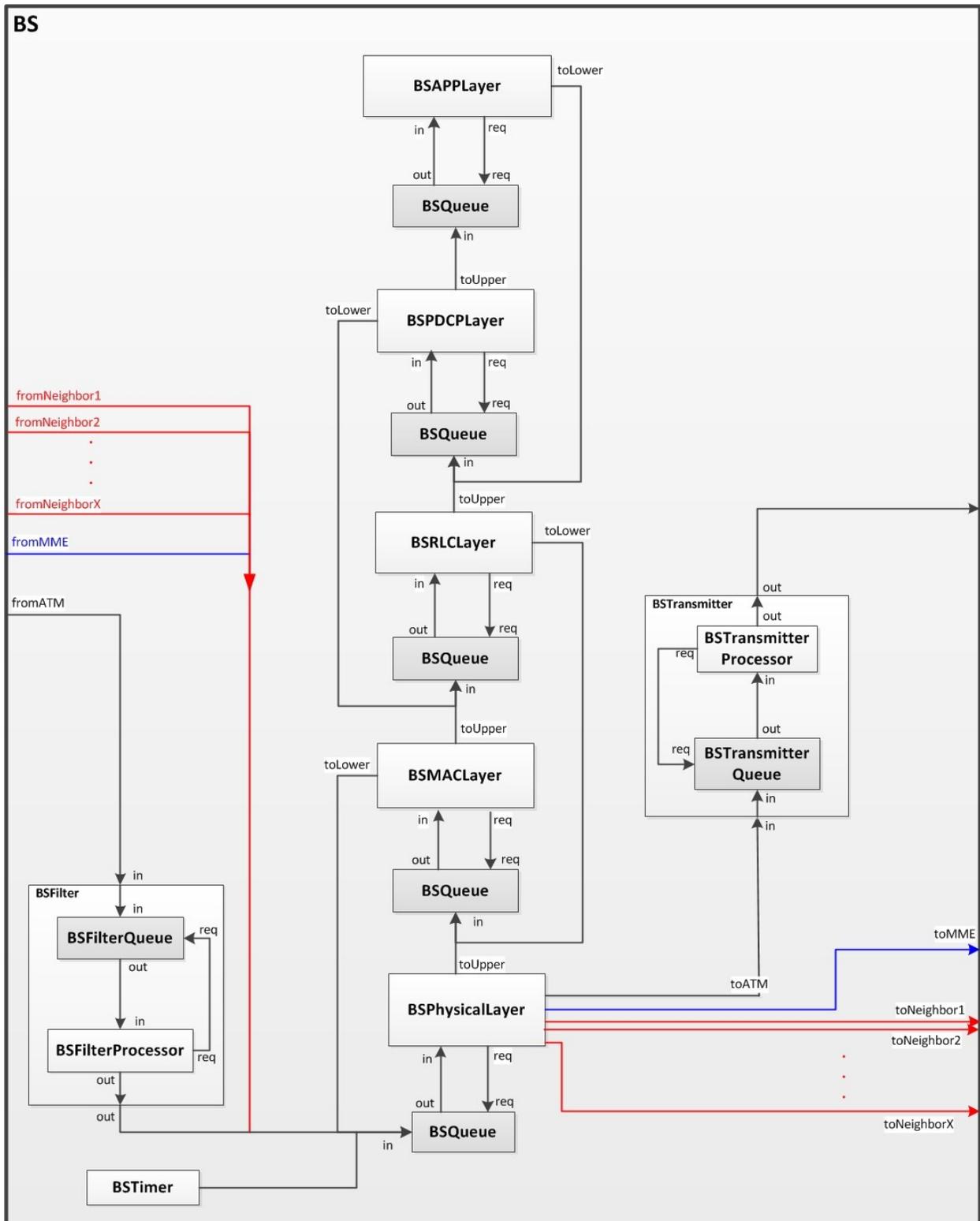


Figure 4.11: Structure of the *BS* coupled model showing atomic and coupled submodels.

Following the DEVS guidelines, the atomic submodels in Figure 4.8, Figure 4.9 and Figure 4.10 are each defined in a separate C++ class, which extend the basic DEVS *Atomic* class. These C++ classes define the implementations of the internal, external, and time advance functions, according to the DEVS formal definition. Figure 4.12 shows the Unified Modelling Language (UML) class diagram for the model's atomic classes.

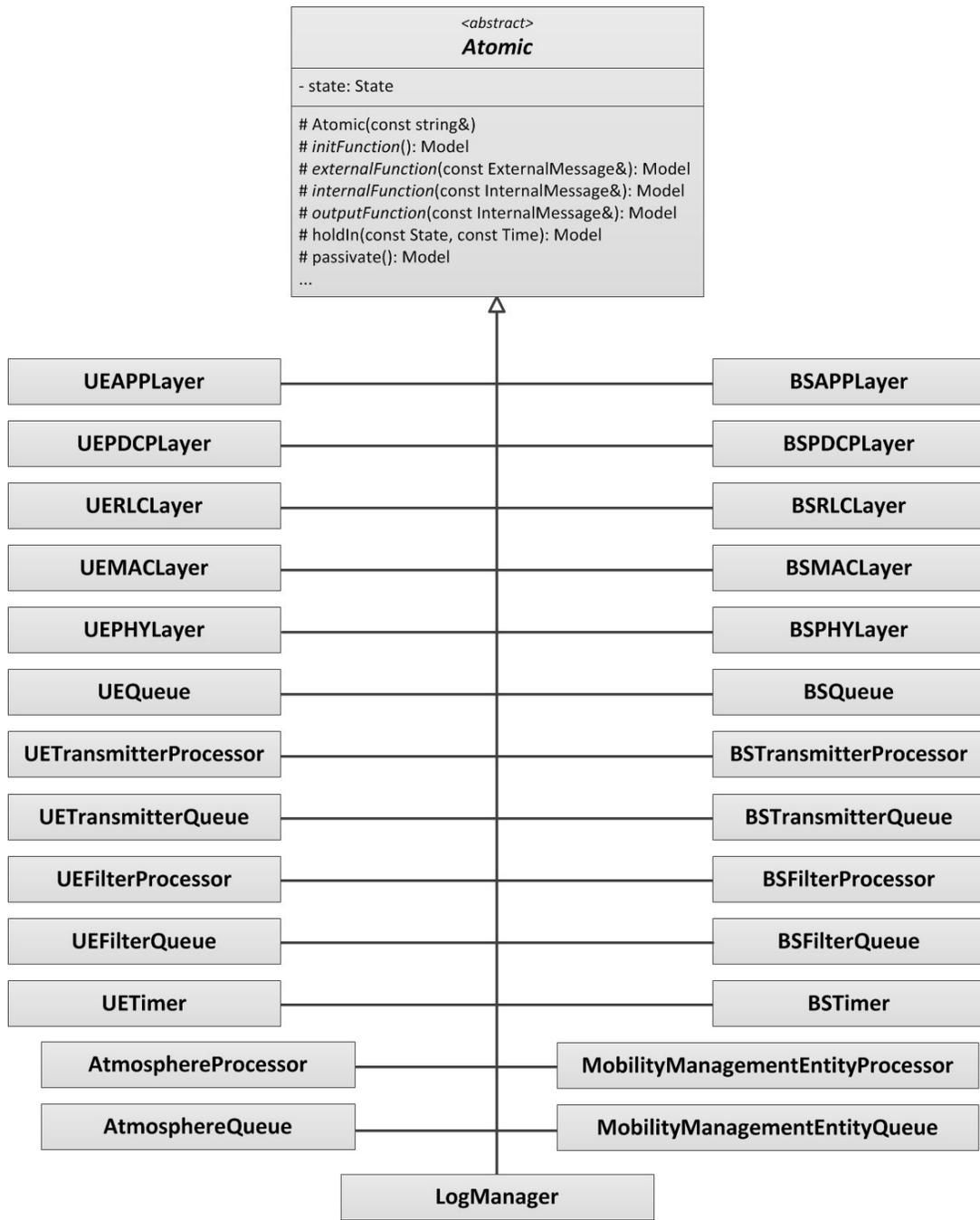
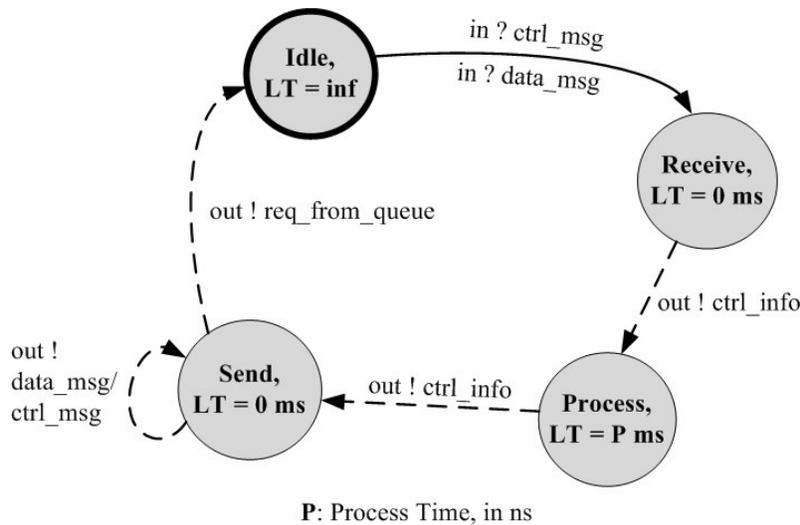


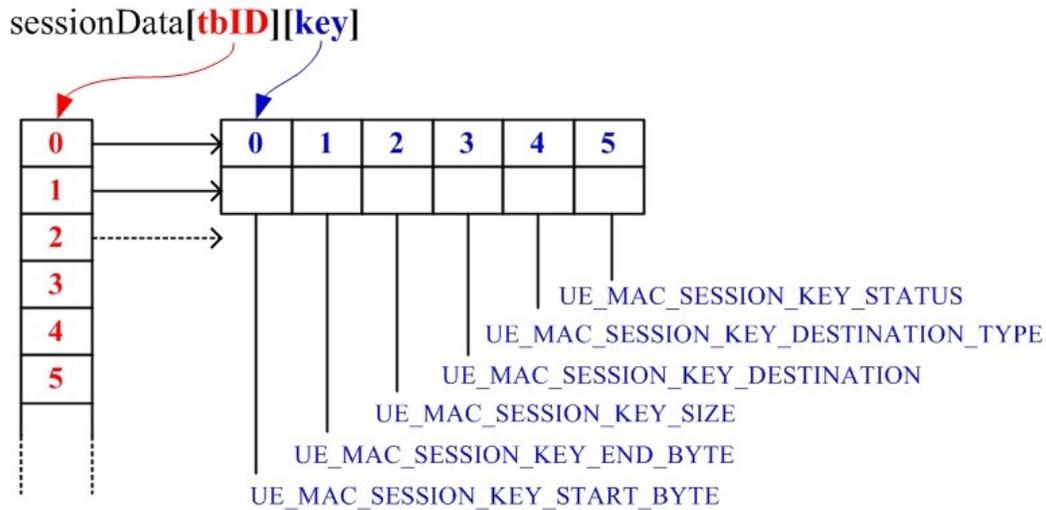
Figure 4.12: Simplified UML class diagram for the model's atomic classes.

To allow the model to be easily adapted to any simulation scenario, the model was designed to be flexible. To do this, the basic functions of LTE systems were implemented separately from the behaviour of the algorithm under consideration. Each of the five layers in Figure 4.10 and Figure 4.11 was designed to handle any number of “communication sessions” with other nodes (UEs or eNBs). Upon receiving a control or data message, a layer can retrieve the corresponding session and act according to the state of that session. Each layer in the model (UEs, eNBs, or MME) receives the messages in order and processes them, according to the definition of the algorithm. For instance, the DEVS graph of the *UEPDCPLayer* is shown in Figure 4.13. As seen in the figure, *UEPDCPLayer* cycles through four states: *Idle*, *Receive*, *Process*, and *Send*. *Idle* is the initial state of the model. When a message is received, the model transitions to the *Receive* state, where the corresponding session is retrieved (or created and stored). The model then transitions to the *Process* state, where the incoming message is processed based on the algorithm definition, and a list of zero or more outgoing data/control messages is created. The model then transitions to the *Send* state and remains in that state while it sends the outgoing messages, one at a time. At the end of the cycle, *UEPDCPLayer* sends a message to the *Queue* model to request the next incoming message and transitions back to the initial *Idle* state. The other atomic classes that model the LTE layers, as well as the application layer, in both the *UE* and *BS* models, have identical states and state transitions as in Figure 4.13.



**Figure 4.13: DEVS graph for *UEPDCPLayer*.**

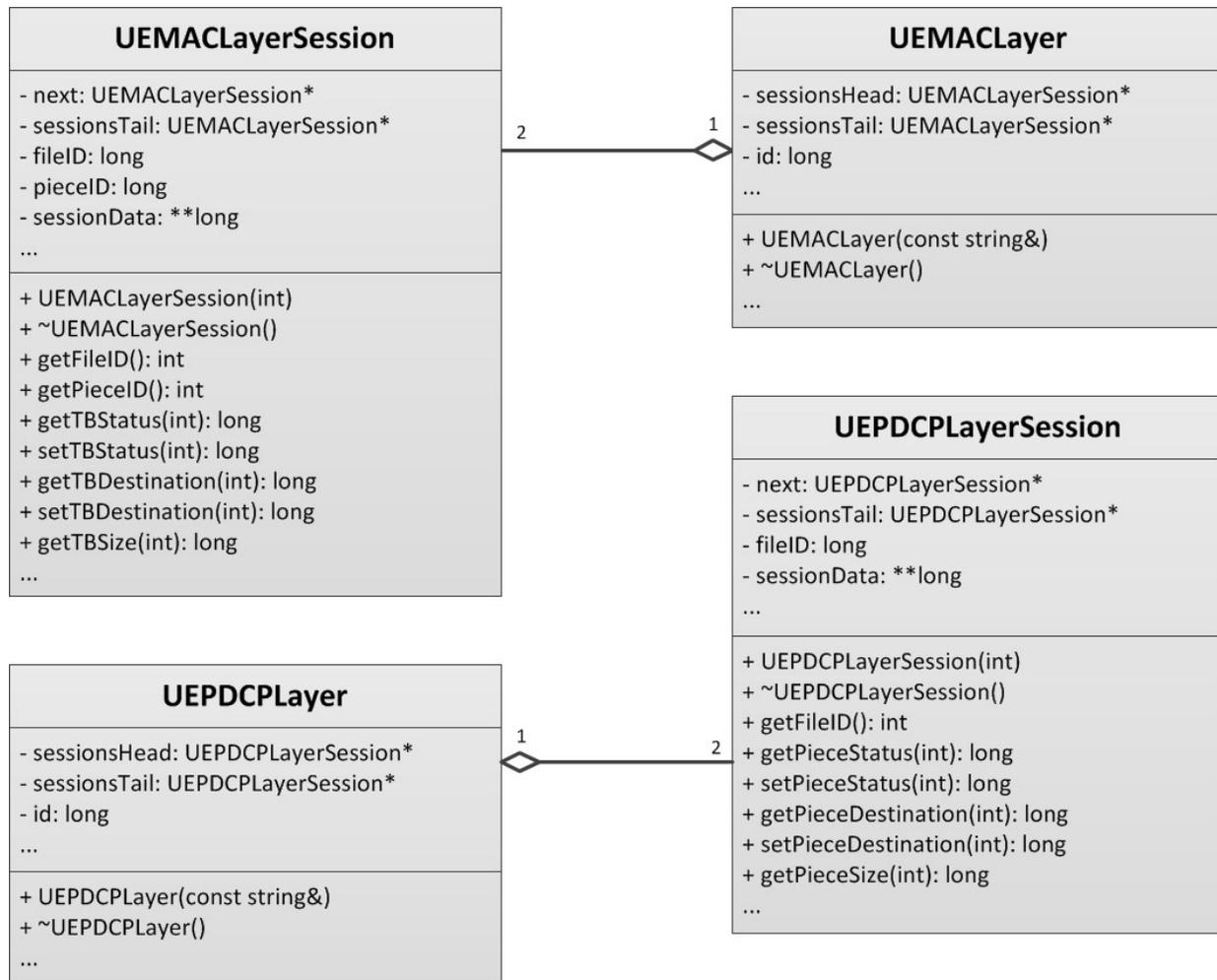
The state of the communication sessions is maintained by the corresponding layer using *Session* classes. Each layer holds a reference to the head and tail of a linked list of *Session* instances. Each of the objects in these lists encapsulates the required information to determine the appropriate action at each stage of the transmission process. For instance, when the *UEMACLayer* receives a file piece from the upper layers, it creates a new *UEMACLayerSession* to store the relevant state information. This information includes the original file piece, a list of TBs that have been created from that piece, the transmission status of each of the TBs, a list of eNBs and helper UEs participating in the upload process, etc. The list of TBs and their transmission status are stored in the *sessionData* variable. The *sessionData* variable is two-dimensional array (or array of arrays) of long integers. The resulting data structure is shown in Figure 4.14. The first index refers to the ID of the TB, while the second index refers to one of a set of keys to retrieve the corresponding status information. For example, *sessionData*[3][UE\_MAC\_SESSION\_KEY\_SIZE] refers to the cell storing the size of the TB with ID 3.



**Figure 4.14: Format of the *sessionData* data structure.**

Similarly, when the *UEPDCPLayer* receives a request to upload a file, it creates a *UEPDCPLayerSession* object to store the file, along with the relevant information about the pieces created and their transmission status, etc. Figure 4.15 shows the structure of the

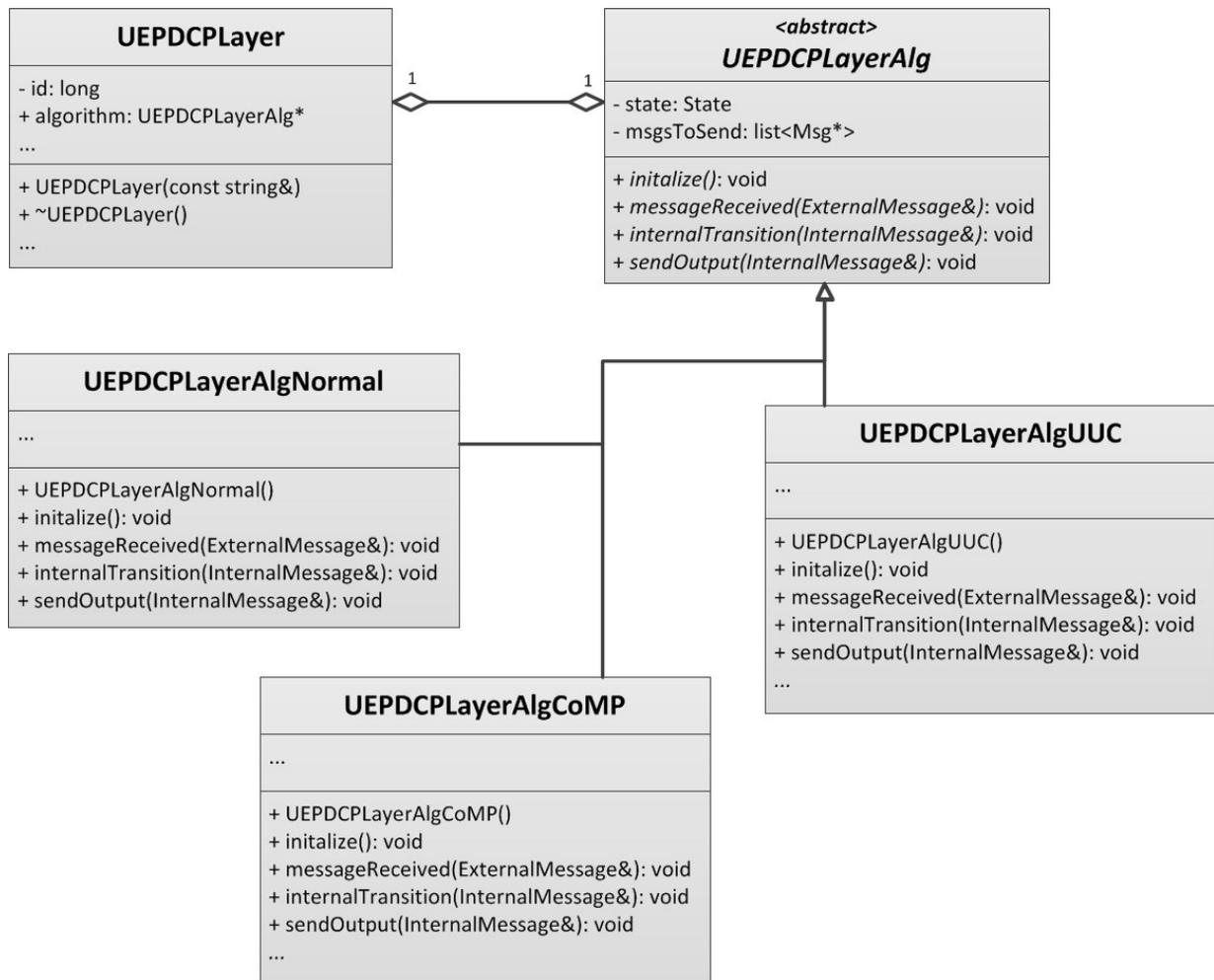
*UEMACLayerSession* and *UEPDCPLayerSession* classes. The session classes for the other layers and the equivalent session classes for a BS are defined based on the same structure.



**Figure 4.15: UML class diagram showing the session classes.**

Moreover, in order to decouple LTE’s basic functions from the definition of the algorithm, a strategy design pattern was used. The pattern allows the definition of a set of algorithms that inherit their basic functions from a common abstract class (called superclass). The definition of each algorithm is encapsulated in a separate class (called subclass), which provides algorithm-specific implementations to these basic functions. The superclass is used to define a common interface to the derived algorithm subclasses. This approach is illustrated in Figure 4.16. The *UEPDCPLayer* class holds a reference to an abstract *UEPDCPLayerAlg* object. At run time, this reference can be assigned any of the concrete algorithm

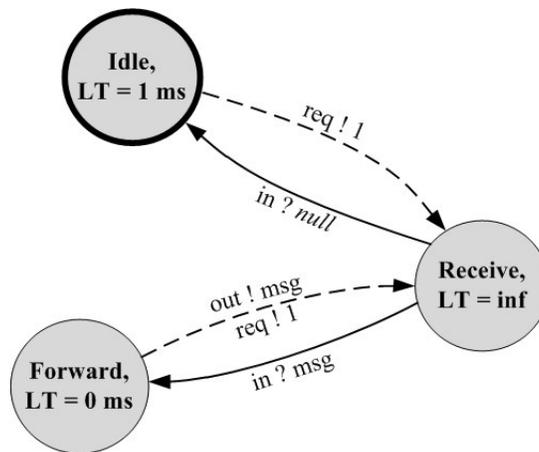
implementations, and is selected based on parameter defined in the model file. In the *Process* state, the *UEPDCPLayer* can pass the incoming message to the algorithm object being used, which is responsible for performing the required operations based on the definition of that algorithm. This approach allows us to select an algorithm for each node in the network independently, and allows the algorithms to be swapped during runtime. Moreover, additional algorithms for the UE's PDCP layer can be easily defined by extending the *UEPDCPLayerAlg* superclass, without altering the layer's definition of the basic LTE functions. The algorithm classes for the other layers (for both UEs and eNBs) follow the same design pattern.



**Figure 4.16: UML class diagram showing the algorithm class hierarchy.**

As mentioned earlier, the *UETransmitterProcessor* and *BSTransmitterProcessor* models are responsible for queuing the outgoing TBs as well as synchronizing the transmission of these

TBs with LTE's TTI. The DEVS graph for *UETransmitterProcessor* is shown in Figure 4.17. The model has three states: *Idle*, *Receive*, and *Forward*. It starts in the *Idle* state, and at the beginning of the first TTI, it sends a request to the queue and immediately transitions to the *Receive* state. It remains in the *Receive* state until it receives a message from the queue. If the queue is empty, i.e., the UE has no TBs to send, the *UETransmitterProcessor* receives a *null* message from the queue, returns to the *Idle* state and waits for the next TTI. If the *UETransmitterProcessor* receives a message, it transitions to the *Forward* state. In this state, the model forwards the received message. Simultaneously, it sends a request to the queue for the next message, and returns to the receiver state. Therefore, the model alternates between the *Receive* and *Forward* state until the queue is empty, then transitions to the *Idle* state and waits for the beginning of the next TTI. The *BSTransmitterProcessor* model operates in a similar manner.



**Figure 4.17: *UETransmitterProcessor* DEVS graph.**

A *UEFilterProcessor* atomic model has the same three states, and it is based on the same *Idle-Receive-Forward* cycle as the Transmitter Processor models. It begins in the *Idle* state, and immediately requests a next message from the queue and transitions to the *Receive* state. Upon receiving a message, the *UEFilterProcessor* checks the destination address of the incoming message. If the receiving UE coupled model is the intended receiver of the message, the *UEFilterProcessor* moves to the *Forward* state where the message is passed through to *UEPHYLayer*'s queue. On the other hand, if the receiving entity is not the intended receiver, the message is discarded. The model then transitions to the *Idle* state, and repeats the process for

each incoming message. Figure 4.18 illustrates this process using a DEVS graph. The same process is used for the *BSFilterProcessor* atomic model.

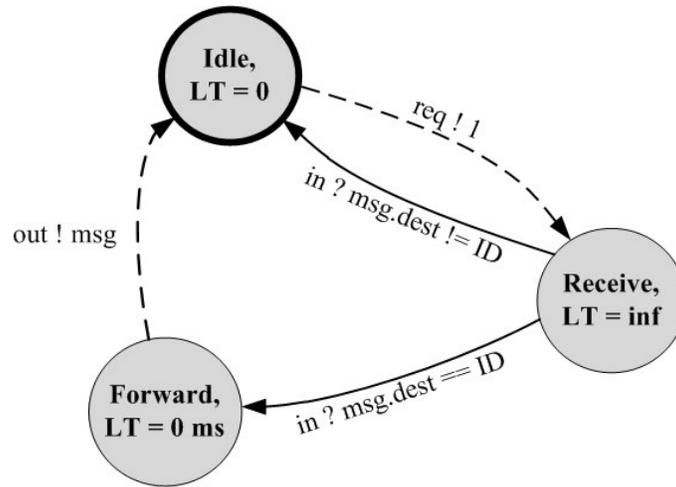


Figure 4.18: *UEFilterProcessor* DEVS graph.

## 4.3 Evaluation of UUC

### 4.3.1 Non-cooperative Algorithm

The non-cooperative algorithm used to evaluate the performance of Shared Segmented Upload (see Section 3.4.1) was modified to work with updated LTE DEVS model. The two non-cooperative algorithms operate in an identical matter. However, in the updated algorithm, the upload process occurs over the five layers described earlier. The upload process starts with an *UploadRequest* message that is sent by the owner UE to its serving eNB, and ends when the serving eNB receives the last piece of the file and sends an *UploadFinished* message back to the owner UE.

### 4.3.2 Standard Coordinated MultiPoint Algorithm

We developed another algorithm to evaluate UUC as well. This algorithm is based on the standard Joint Transmission (JT) Coordinated MultiPoint (CoMP) technique. A UE employing

this algorithm uploads each transport block based on the propagation models discussed earlier. Each eNB participating in the UE's coordination set receives its own copy of the transmitted signal. In other words, the UE sends a single transmission and the signal is received by multiple antennas that are geographically distributed at multiple BS sites. This form of joint reception is usually referred to as receive diversity [67]. The eNBs then share these copies with the serving eNB (over the X2 backhaul links) which, based on the signal-to-noise (SNR) ratio of the received signals, selects the signal from the diversity branch (one of the receiver eNBs) with the highest instantaneous SNR. This approach of selecting the signal with the maximum SNR is referred to as Conventional Selection Combiner (CSC) [68]. Compared to UUC, a UE employing this algorithm only communicates with the eNBs in its coordination set, and does not communicate with neighboring UEs. In uplink JT CoMP, when a UE is sufficiently close to their serving eNB, the signal received by cooperating non-serving eNBs is likely to be weak. The cooperation between eNBs in such cases would not result in any adequate performance gains, but would impose an overhead on the network, which could potentially offset any achievable gains. Therefore, to avoid the extra overhead outweighing the cooperative gain, we take into account the average SNR received by the neighboring eNBs prior to the start of the upload. In this case, non-serving eNBs only join the coordination set if the difference between the average SNR of the received uplink transmissions at the serving and non-serving eNBs, is less than 9 dB [69, 70].

Similar to the non-conventional algorithm, the upload begins when the UE sends an *UploadRequest* message to the eNBs in range, and ends when the UE receives an *UploadFinished* message from its serving eNB.

### 4.3.3 Propagation models and transmission parameters

In the updated LTE DEVS model used to simulate UUC, the same propagation model from the previous model (presented in Chapter 3) is used. The propagation model is defined in Section 3.4.2, in equations (1), (3), (4), (5). The propagation model considered for D2D communication assumes outdoor-to-outdoor communication, i.e., both the transmitting and receiving UEs are located outdoors. Based on the channel models defined in [71], the path loss for a direct Line-of-Sight (LoS) transmission between two UEs can be calculated using equation

(6) [72]. LoS refers to a transmission that occurs when both transmitting and receiving UEs are in view of each other without any obstacles between them. Similarly, the path loss for non-LoS (NLoS) transmissions is calculated using equation (7) [72].

$$L_{D2D, LoS} = (22.7 * \log_{10} d) + 27.0 - (20.0 * \log_{10} f) \quad (6)$$

$$L_{D2D, NLoS} = ((44.9 - 6.55 * \log_{10} h) * \log_{10} d) + 5.83 * \log_{10} h + 14.78 + 34.97 * \log_{10} f \quad (7)$$

where  $d$  is the UE-to-UE separation distance, in meters;  $f$  is the operating carrier frequency for D2D transmissions, in MHz; and  $h$  is the UE antenna height, calculated from the ground level in meters. The NLoS path loss is offset by -5 dB to adapt the channel model for D2D communication.

The probability of having a LoS transmission is used to select one of the two channel models described in equation (6) and equation (7) to calculate the overall channel path loss ( $pathloss_{B1}$ ). The probability is calculated using equation (8) [73].

$$P_{LoS} = \left( \min\left(\frac{18}{d}, 1\right) * \left(1 - \exp\left(-\frac{d}{36}\right)\right) \right) + \exp\left(-\frac{d}{36}\right) \quad (8)$$

The channel path loss is then calculated based on equation (9) [71].

$$pathloss = \max(pathloss_{freespace}, pathloss_{B1}) \quad (9)$$

where  $pathloss_{freespace}$  is the free space path loss (or the minimum path loss a channel may experience) and is calculated using equation (10) [73].

$$pathloss_{freespace} = (20 * \log_{10} d) + 46.4 + \left(20 * \log_{10} \left(\frac{f}{5.0}\right)\right) \quad (10)$$

Shadowing was implemented using a log-normally distributed random variable,  $D2DLogF$ . Moreover, a Rayleigh distributed random variable ( $rayleighFading$ ) was used to model small-scale fading. The resulting propagation losses are combined using equation (11).

$$pathlossPlusFading = L + Log + rayleighFading \quad (11)$$

Finally, the received power and data rate of the transmission link are calculated using equation (12) and equation (13) respectively. Equation (13) takes into account additive white Gaussian noise [64].

$$Pwr_R = Pwr_T - \text{Max}(\text{pathlossPlusFading} - G_T - G_R, MCL) \quad (12)$$

$$\text{data rate} = B \log_2\left(1 + \frac{Pwr_R}{N_0 \times B}\right) \quad (13)$$

The simulations in this model were carried out in an urban area setting with an operating carrier frequency of 900 MHz. However, all D2D communications operate on a carrier frequency of 2000 MHz [71]. 5 MHz was used as the transmission channel bandwidth. A fixed noise density was set to -174 dBm/Hz for all communication channels. The log-normal shadowing used in the uplink and downlink transmissions was calculated using a standard deviation of 8 dB. However, the shadowing for D2D transmissions was calculated using a standard deviation of 7 dB. The eNB antenna height was set to 15 meters above the average rooftop height (30 meters from ground level), while the UE antenna height was fixed to 1.5 meters. The common transmission parameters are summarized in Table 4.1, while the remaining specific transmission parameters used for LTE uplink/downlink and D2D channels are summarized in Table 4.2 and Table 4.3, respectively.

| Parameter                     | Value       |
|-------------------------------|-------------|
| Transmission bandwidth        | 5 MHz       |
| Noise Density                 | -174 dBm/Hz |
| Dhb                           | 15 meters   |
| Hb                            | 45 meters   |
| Noise figure (eNB)            | 5 dB        |
| Maximum UE transmission power | 21 dBm      |
| Minimum UE transmission power | -50 dBm     |
| Noise figure (UE)             | 9 dB        |

**Table 4.1: Common transmission parameters.**

| Parameter                    | Value   |
|------------------------------|---------|
| Carrier frequency            | 900 MHz |
| LogF                         | 8 dB    |
| Maximum eNB power            | 43 dBm  |
| Maximum power per DL channel | 30 dBm  |
| Minimum eNB power per UE     | 15 dBm  |

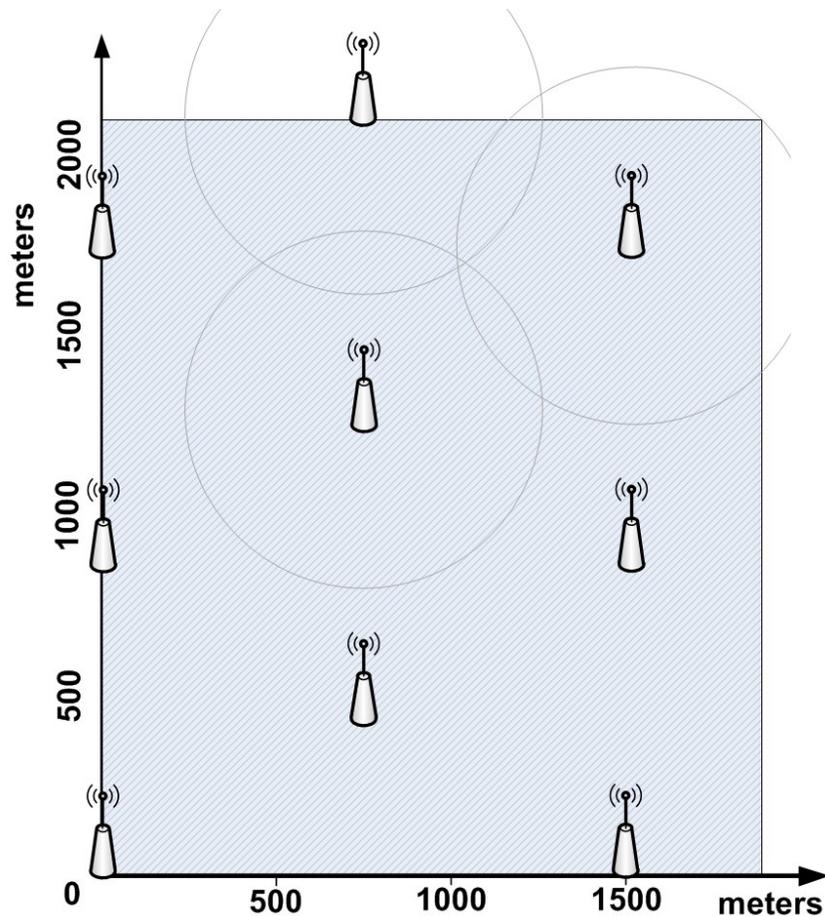
**Table 4.2: Transmission parameters for uplink/downlink transmissions.**

| Parameter                 | Value      |
|---------------------------|------------|
| Carrier frequency         | 2000 MHz   |
| D2DLogF                   | 7 dB       |
| Average UE Antenna Height | 1.5 meters |
| NLoS offset               | -5 dB      |

**Table 4.3: Transmission parameters for D2D transmissions.**

#### 4.3.4 Simulation scenarios and results

We conducted a number of system-level simulations to evaluate the potential performance gains of UUC, and compare it to the performance of a conventional non-cooperative upload, as well as a standard CoMP upload. Two simulation scenarios and the collected results are presented in this section. Both simulations scenarios consider an urban area of 1875 meters by 2165 meters, served by nine eNBs. Each eNB covers a cell area with a 500-meter radius. The network topology is shown in Figure 4.19.



**Figure 4.19: Network topology showing eNB positions.**

In each simulation that utilizes UUC, we compare the upload performance of owner UEs using the conventional non-cooperative algorithm, the standard CoMP algorithm, as well as UUC with varying number of helper UEs. The model was set up in such a way that allows us to preselect the number of helper UEs that are willing to assist each owner UE. This allows the required simulation scenario to be configured using the model file. Moreover, the positions of the UEs and the owner UE to helper UE separation distances can also be configured in the model file.

In the first simulation scenario, we compare the performance of the three algorithms as a function of the distance between the UEs and their serving eNBs. This scenario is similar to the first scenario presented in Chapter 3 (Figure 3.13). The BS-UE separation distances are increased in each simulation run in increments of 50 meters. However, in each simulation run, the UEs are

placed within a distance range of 10 meters, starting with 100-110 meters. For example, in the first simulation run, the UEs are placed between 100 and 110 meters from their serving eNBs. In the second simulation run, the UEs are placed 150 to 160 meters away from their serving eNBs, and so on. In the final round of simulations, the UEs are placed within a distance range of 450-460 meters from the cell center.

In this scenario, we simulate the upload of ten data files by ten owner UEs. In the UUC simulations, the owner UE to helper UE separation distance is fixed to 10 meters. This was done in order to reduce the variability of the results caused by varying UE-to-UE distance. The size of the data files to be uploaded by the owner UEs are selected based on a uniform distribution between 30 and 60 Megabytes. A uniform piece size of 1 Megabyte was selected. The simulation parameters for scenario 1 are summarized in Table 4.4.

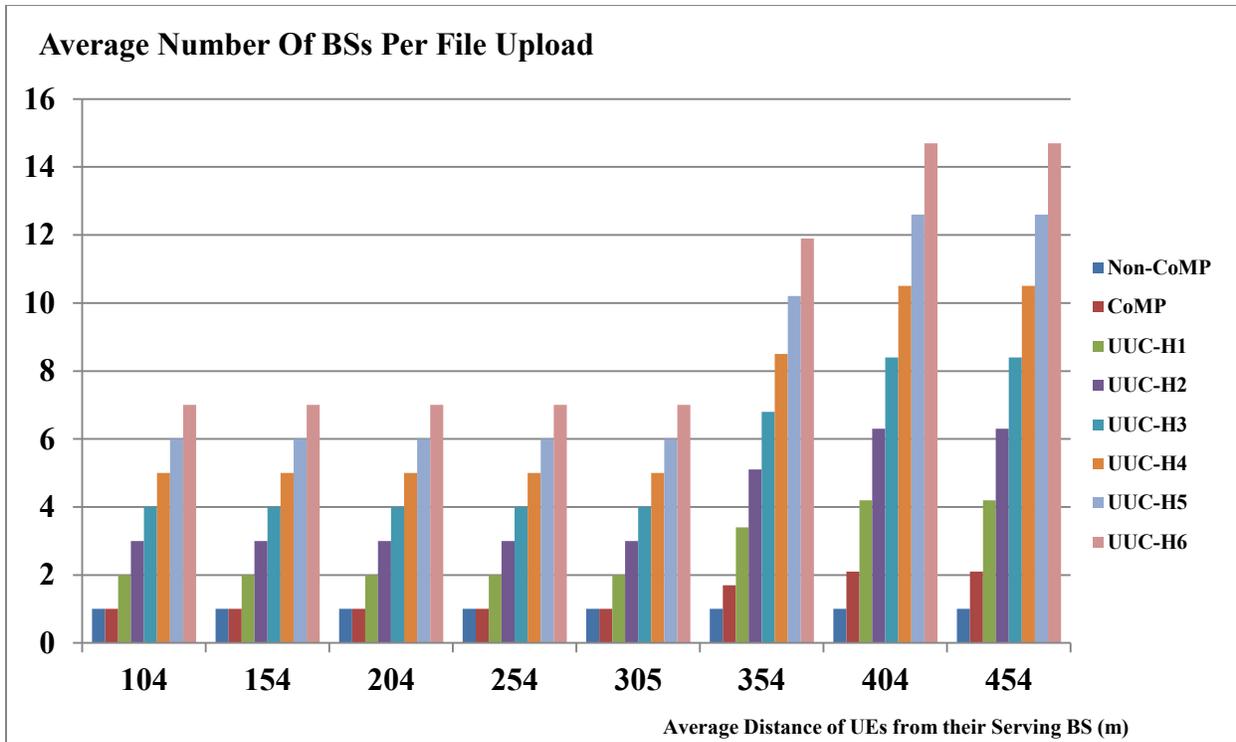
| Parameter                      | Value              |
|--------------------------------|--------------------|
| Owner UE to Helper UE distance | 10 meters          |
| UE to serving eNB distance     | Variable           |
| File Size                      | 30 to 60 Megabytes |
| Piece Size                     | 1 Megabyte         |

**Table 4.4: Parameters for simulation scenario 1.**

For each UE-to-BS distance range, eight simulations were conducted as follows. In the first simulation, each owner UE only uploads the file to its serving eNBs using the traditional non-cooperative approach presented in section 4.3.1. We will refer to this setup as “Non-CoMP”. In the second simulation (referred to as “CoMP”), each owner UE uploads their data file to its coordination set of eNBs using the standard CoMP algorithm described in Section 4.3.2. Note that the availability of multiple cooperating eNBs depends on the position of the UE within the cell, as well as the channel quality. In the following six simulations, the owner UEs upload their files using UUC with the help of one to six helper UEs, respectively. We will refer to these simulations as “UUC-H1”, “UUC-H2”, and so on. The simulations were allowed to run until all the files were uploaded by the UEs. The results of simulation scenario 1 are shown in Figure 4.19, Figure 4.20, and Figure 4.21.

Figure 4.20 shows the average number of eNBs that participate in the upload of files (per file) versus the average distance of UEs from their serving eNBs. During the simulation, the owner UEs are able to upload file pieces to the eNBs of their coordination set. Moreover, each helper UE participating in the upload process can upload pieces of the same file to the eNBs in their own coordination set. The number of eNBs in Figure 4.20 includes the eNBs that received file pieces from owner UEs as well as helper UEs (even if the eNBs in the coordination sets of the owner and helper UEs overlap). For example, consider an owner UE uploading a file with the help of one helper UE. If the owner and helper UE share the same coordination set of three eNBs, the average total number of eNBs for that file would be six. In other words, the average number of BSs per file represents the number of uplink transmission channels that have been used to support the file upload. From the figure, it can be seen that while the UEs are within 305 meters or less from their serving eNBs, Non-CoMP and CoMP behave in an identical manner. This is expected, since non-serving eNBs only join the coordination set for a UE if the average received SNR is within 9 dB of the serving eNB's average received SNR, and therefore, the UE has to be relatively close to the cell edges to communicate with multiple eNBs. At the cell edges, the UEs employing CoMP are able to upload pieces of their files to 2.1 eNBs, on average.

Close to the cell center, each UE only communicates with its serving eNB. The addition of a helper UE to the upload process increases the average number of BSs by one. However, at the cell edges, UEs are able to communicate with multiple eNBs. In the case of UUC-H6, where the upload of each file is assisted by six helper UEs, the file is uploaded through 14.7 transmission channels, compared to 2.1 for UEs employing CoMP, and 1 for UEs using the traditional non-CoMP approach. Increasing the number of transmission channels that can be utilized in parallel for a file upload offers an opportunity to improve the overall performance for file uploads.



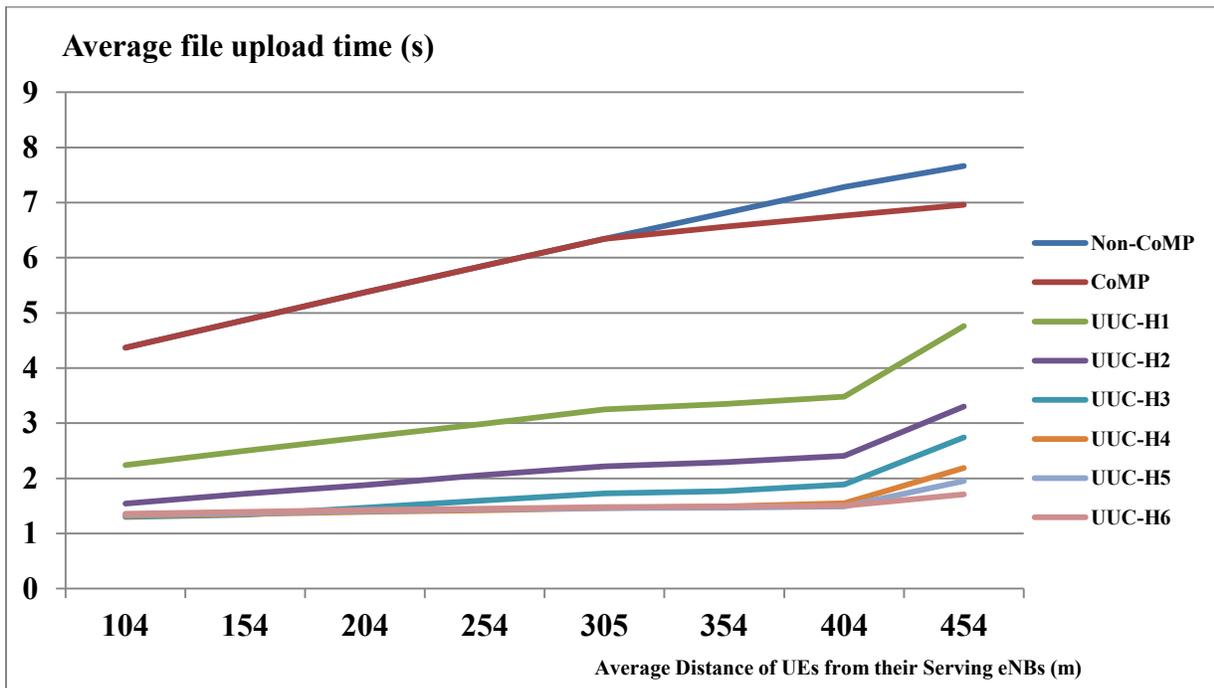
**Figure 4.20: Average number of BSs per file vs. average distance from cell center.**

Figure 4.21 shows the average upload time per file versus the average distance of UEs from their serving eNBs. As previously discussed, Non-CoMP and CoMP behave in a similar manner when the UE is within 305 meters from the cell center. As the UEs move towards the cell edges, the average upload time increases steadily, as the quality of the transmission channels between the UEs and their serving eNBs is reduced. Closer to the cell edge, it can be seen that UEs employing CoMP are able to upload their files in less time, compared to UEs using the non-cooperative approach. At the cell edges, CoMP reduces the average upload time from 7.66 seconds to 6.96 seconds (approximately a 9.1% reduction).

When UEs employing UUC are near the center of the cell, the addition of one helper UE to the upload process reduces the upload time by approximately half. At the cell edges, the performance gain achieved by the additional helper UE is reduced due to the overhead that CoMP imposes on the network. The performance gain achieved with the addition of each subsequent helper UE is further reduced, since the data files are shared with more helper UEs. For example, when the UEs are located around 104 meters away from their serving eNBs, the addition of a third helper UE reduces the average upload time by 15.6% (from 1.54 seconds to

1.3 seconds), compared to the 31.2% reduction (from 2.24 seconds to 1.54 seconds) by the addition of a second helper UE.

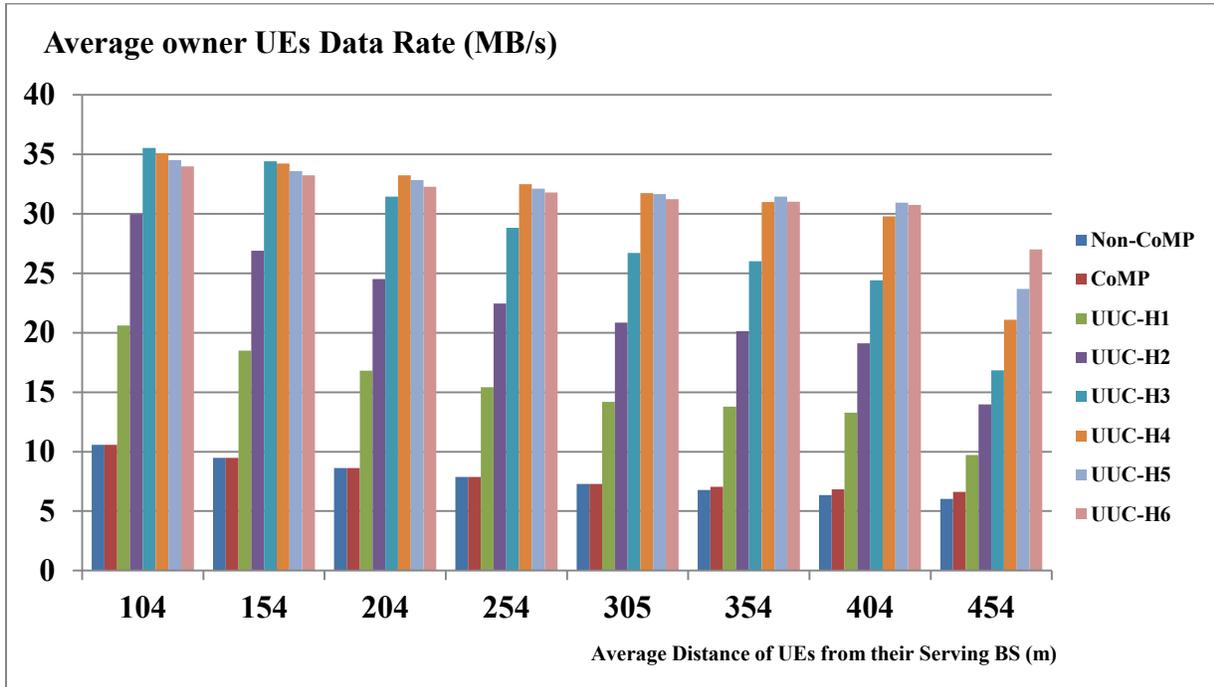
Near the cell center, the quality of the transmission channel between the owner UE and its serving eNB is relatively good. The performance gain achieved by the addition of a fourth helper UE is outweighed by the overhead of UUC. This overhead is caused by the additional transmission of data between the owner UE and the helper UE. However, when the UEs are at the cell edges, performance gains can still be achieved with the assistance of six helper UEs.



**Figure 4.21: Average upload time vs. average distance from serving eNB.**

Figure 4.22 illustrates the same results with respect to the average data rate per file upload versus the average distance of the UEs from their serving eNBs. The average data rate is calculated per file, by dividing the size of the file by the total amount of time spent uploading the file. The resulting data rate at which the file is uploaded increases with each additional helper UE, when the UEs are close to the cell center. However, increasing the number of helper UEs beyond three reduces the average data rate as it increases the overhead on the network. At the cell edges, where the quality of the transmission channel between the owner UE and its serving eNB is poor, the assistance of six helper UEs to the upload process increases the average data

rate from 6.63 Megabytes/second (CoMP at 454 meters) to 27.01 Megabytes/second (UUC-H6 at 454 meters).



**Figure 4.22: Average data rate of file upload vs. average distance from serving eNB.**

As previously mentioned, the first simulation scenario was conducted using uniformly distributed file sizes, between 30 and 60 Megabytes (MB), while varying the distance between UEs and their serving eNBs. The results revealed that the helper UEs offer significant performance gains to the owner UEs. However, the addition of four or more helper UEs offer no additional performance gains when the owner UE is close to the serving eNB.

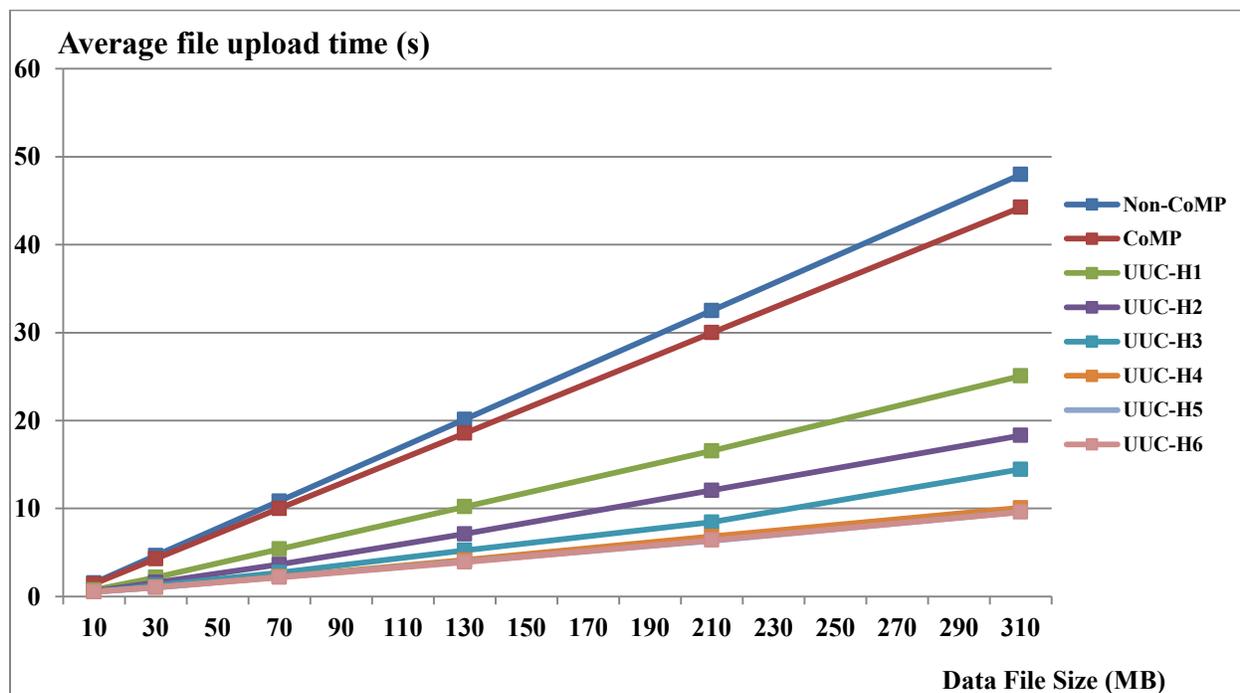
The achievable performance gains also depend on the size of the data file. A very small file would incur a large amount of overhead to set up the D2D communication channels between the owner UE and helper UEs. This would reduce the upload performance of the owner UE. On the other hand, a relatively large file would make the overhead of setting up the D2D communication channels insignificant compared to the overall amount of time required to upload the file. To explore the effect of the data file size on the achievable performance of UUC, we conducted a second set of simulations. In this scenario, we focus on cell-edge UEs which are placed randomly between 350 meters and 500 meters from their serving eNBs. The owner UE to

helper UE distance is kept fixed at 10 meters. The upload performance of UEs for each of the methods presented was investigated using the following file sizes: 10 MB, 30 MB, 70 MB, 130 MB, 210 MB, and 310 MB. The remaining simulation parameters for the second scenario are listed in Table 4.5.

| Parameter                      | Value             |
|--------------------------------|-------------------|
| Owner UE to Helper UE distance | 10 meters         |
| UE to serving eNB distance     | 350 to 500 meters |
| File Size                      | Variable          |
| Piece Size                     | 1 megabyte        |

**Table 4.5: Parameters for simulation scenario 2.**

Figure 4.23 shows the average file upload time versus the file size for UEs employing each of the presented upload algorithms. The figure shows that the average time required to upload files increases linearly as the size of the files increases. CoMP offers between 7% and 8% reduction in the upload time, regardless of the file size, compared to the standard non-cooperative method.



**Figure 4.23: Average upload time per file vs. file size.**

UUC further improves the upload performance for cell-edge UEs. The achievable performance improvement gained by using UUC depends on the availability of helper UEs. For example, owner UEs that use UUC with the assistance of one helper UE obtain an average of 46.1% reduction in the upload time, compared to UEs that utilize the standard CoMP approach. Similarly, the assistance of a second helper UE increases the upload time reduction to an average of 60.7% compared to CoMP. Increasing the number of helper UEs to four helper UEs further improves the effectiveness of UUC. However, the addition of the fourth helper UE only improves the upload performance when the data files are sufficiently large. For small files in the order of 10 to 20 Megabytes, increasing the number of helper UEs beyond three UEs has little to no effect on the upload performance, and, in some cases, the overhead of UUC outweighs the gains and increases the overall upload time. For files that are 30 Megabytes or larger, the addition of the fourth helper UE further reduces the upload time (77.1% reduction on average, compared to CoMP). Beyond four helper UEs, the assistance of additional helper UEs provides insignificant improvement in the upload time. In some cases, these additional helper UEs impose an overhead that is larger than the performance gains they provide, resulting in a lower overall performance. These results are clearly illustrated in Figure 4.24. Figure 4.24 shows the average data rate per file upload versus the file size. It can be seen from this figure that there are significant improvements to the average data rate when each owner UE is assisted by up to three helper UEs. For files smaller than 30 Megabytes, there is no significant effect on the data rate with each additional helper UE, beyond three helper UEs. On the other hand, for files larger that are 30 Megabytes or larger, a fourth helper UE provides further improvements in the upload performance, and increases the average data rate. Beyond four helper UEs, additional helper UEs has an insignificant effect on the average data rate, especially for small sized files.

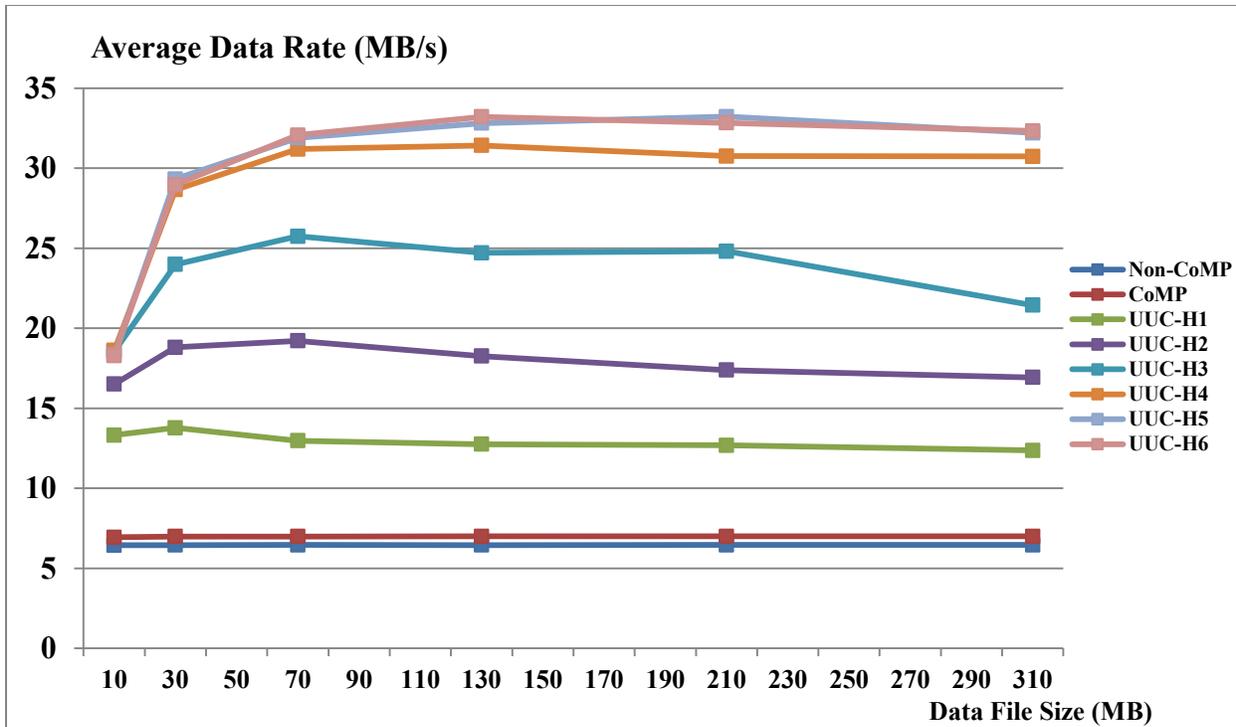


Figure 4.24: Average data rate per file vs. file size.

## 4.4 Model Verification and Validation

An important step in the development of a simulation model is Verification and Validation (V&V). Model V&V are the primary methods used to assess the accuracy and reliability of the model. Verification is the process of confirming that the computer model has been correctly implemented based on the specifications of the conceptual model. The objective of the process is to insure that the implementation of the model generates the expected output for a given set of inputs. On the other hand, validation deals with the accuracy of the conceptual model itself, as well as how closely it represents the system being modelled.

As previously mentioned, the DEVS model presented in this chapter was developed in collaboration with experts from Ericsson Canada. This collaboration has helped us with the process of verifying and validating our LTE model.

In terms of model validation, the conceptual model along with the assumptions and simplifications made were discussed with Ericsson Canada at each stage of the development

process. These discussions helped refine the conceptual model into one that was deemed reasonably accurate for the simulations presented in this chapter.

With the help of Ericsson Canada, the model's implementation was also verified. Since DEVS models are modular in nature, each component (or submodel) was individually tested using the black-box software testing approach. Using this approach, each component was given different set of inputs and the generated output was compared to the expected output. For instance, one test case for the *UEPHYLayer* model was to send a TB message to the input port of the *UEPHYLayer* model. The model, as expected, produced an outgoing TB message from the correct output port to the upper layers, with the expected header information and processing delay. Once each component is verified individually for each possible set of inputs, the components are integrated into a larger model, which is then verified again in a similar manner.

## 4.5 Summary

In this chapter, we presented the Upload User Collaboration (UUC) algorithm to improve the upload performance of UEs trying to upload large files to a cellular LTE-A network. The algorithm allows these UEs (called owner UEs) to request assistance from neighbouring UEs (called helper UEs). The owner UEs divide the file into a number of pieces and shares the file pieces with neighboring UEs over Device-to-Device (D2D) communication links. The owner and helper UEs can then upload pieces of the file to the network simultaneously, on separate transmission channels. Moreover, each of these UEs employ Coordinated MultiPoint Joint Transmission techniques to communicate with multiple neighboring eNBs when the UEs are near the cell edges. In this chapter, we also presented a DEVS model for LTE networks that was used to simulate and compare the proposed algorithm to the traditional non-cooperative method, as well as the standard CoMP approach. The simulation results show that UUC promises significant improvements in the upload performance of UEs, regardless of their position within the cell, compared to the conventional approaches (non-CoMP and CoMP). Moreover, we have also shown that significant performance gains can be achieved by UUC for any file size, from 10 to 310 Megabytes.

## Chapter 5

# Conclusion and Future Work

The continually increasing demand for mobile data traffic is presenting new challenges on cellular network deployments. With the evolution of mobile networks as well as the popularity of smartphones, new multimedia applications have recently introduced more strict Quality-of-Service requirements on the cellular networks. Moreover, the availability of usable radio spectrum is scarce. To address these challenges, service providers introduce new communication techniques and algorithms that aim to improve the spectral efficiency of their networks.

There is a strong need to evaluate the performance and efficiency of new techniques and algorithms before they are standardized. As these algorithms become more complex, it becomes unsuitable to test them on live networks. Modeling and simulation techniques allow researchers to model networks and perform simulations to evaluate the performance of the proposed algorithms. This process reduces the cost and time required to develop new protocols. Discrete Event System Specification (DEVS) is a general and formal framework for modeling and simulation that allows the researchers to model complex systems without worrying about the implementation details of the simulator.

In this dissertation, we presented a DEVS-based model for simulating LTE networks. The model was designed to be extensible, flexible and easily adaptable for simulating proposed protocols. The model implements the basic functions of each of the layers of the LTE protocol

stack for UEs, eNBs and the MME. The model's structure allows the modeller to define the behaviour of the algorithm being tested and evaluate its performance at each layer.

In addition, we also presented two algorithms, Shared Segmented Upload (SSU) and Upload User Collaboration (UUC), which have been developed in collaboration with fellow students from Carleton University, and Ericsson Canada. Both algorithms utilize LTE's CoMP techniques and aim to improve the performance of file uploads for users of an LTE network, especially for cell edge users. SSU works by splitting the data file into a number of pieces and uploads the pieces of the file to the eNBs in the UE's CoMP coordination set, simultaneously. UUC works by utilizing the upload power of multiple UEs in close proximity to enhance the upload process of the owner UE. Similar to SSU, a UE employing UUC uploads a data file in a series of pieces to the eNB in the coordination set. However, in addition to the UE-to-eNB upload, the owner UE requests assistance from neighbouring helper UEs with the upload process. The owner UE and helper UEs can then upload pieces of the files in parallel.

The DEVS LTE model presented in this thesis was used to model and simulate the proposed algorithms and compare them to the traditional non-cooperative approach as well as the standard CoMP method. The simulation results show that both algorithms offer significant improvement in the upload performance for UEs who wish to upload data files to their cellular networks. Moreover, SSU provides services that are more consistent to users as their distance increases from their serving eNBs. UUC offers similar improvements; however, it also offers significant improvements for users at or near the center of their cells. Therefore, UUC can also be applied to situations where coordination between eNBs through CoMP is not available.

The simulation scenarios presented in Chapter 4 of this dissertation were used to evaluate the performance of UUC. These simulation scenarios and the results obtained from the simulations show that UUC provides substantial improvements to the performance of users who wish to upload large files to the network. However, these simulation scenarios focus on evaluating the effect of two variables, the distance between the UEs and their serving eNBs, and the size of the files, on the effectiveness of UUC. Other variables include the separation distances between the owner UEs and the helper UEs, the size of the pieces, and the number of

pieces that the owner UE assigns to each of the helper UEs, etc. The effects of these parameters on the performance of UUC should be investigated.

While the model presented in this dissertation represents a good start in modeling LTE networks using Discrete Event System Specification, it models only a relatively small subset of the technologies used in LTE networks, and thus, future work is required to expand the library of available models. For instance, the different scheduling techniques used in LTE (Round-robin, proportional-fair, and Max C/I [35]) can be implemented. Similarly, different coding and modulation schemes, propagation models, and retransmission techniques can also be added. The implementation of these extensions would also make use of the design pattern presented in Chapter 4 allowing a modeller to select the desired scheme or technique through a parameter in the configuration model file.

A possible extension to the proposed Upload User Collaboration algorithm is the definition of the required control messages between eNBs to allow handoff. Handoff occurs when an active UE in an ongoing communication session moves across cell boundaries. It allows the communication session to “handed off” from one serving eNB to another non-serving eNB without interrupting the data flow. Likewise, another possible extension to UUC is to allow owner UEs to upload files while they are out of the network coverage area by using neighboring UEs (who are within the coverage area) as relays.

## References

- [1] Ericsson, “Ericsson Mobility Report - November 2015,” 2015. [Online]. Available: <http://www.ericsson.com/res/docs/2015/mobility-report/ericsson-mobility-report-nov-2015.pdf>. [Accessed: 15-Aug-2016].
- [2] Kleiner Perkins Caufield Byers, “Internet Trends 2016,” 2016. [Online]. Available: <http://www.kpcb.com/file/2016-internet-trends-report>. [Accessed: 16-Aug-2016].
- [3] Ericsson, “Ericsson Mobility Report - June 2016,” 2016. [Online]. Available: <https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf>.
- [4] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno, “Coordinated multipoint transmission/reception techniques for LTE-advanced [Coordinated and Distributed MIMO],” *IEEE Wirel. Commun.*, vol. 17, no. 3, pp. 26–34, 2010.
- [5] S. Parkvall and D. Astely, “The evolution of LTE towards IMT-advanced,” *J. Commun.*, vol. 4, no. 3, pp. 146–154, 2009.
- [6] M. Tavanpour, G. Wainer, G. Boudreau, and R. Casselman, “DEVS-based modeling of coordinated multipoint techniques for LTE-advanced,” in *Proceedings of the 16th Communications & Networking Symposium*, 2013, p. 13.
- [7] A. Osseiran, J. F. Monserrat, and W. Mohr, *Mobile and wireless communications for IMT-advanced and beyond*. John Wiley & Sons, 2011.
- [8] V. S. Frost, W. W. LaRue, and K. S. Shanmugan, “Efficient techniques for the simulation of computer communications networks,” *IEEE J. Sel. areas Commun.*, vol. 6, no. 1, pp. 146–157, 1988.
- [9] J. Pan and R. Jain, “A survey of network simulation tools: Current status and future developments,” *Washingt. Univ. St. Louis, Tech. Rep*, 2008.
- [10] E. Weingartner, H. Vom Lehn, and K. Wehrle, “A performance comparison of recent network simulators,” in *2009 IEEE International Conference on Communications*, 2009, pp. 1–5.
- [11] G. A. Wainer, *Discrete-event modeling and simulation: a practitioner’s approach*. CRC Press, 2009.
- [12] B. P. Zeigler, “DEVS representation of dynamical systems: Event-based intelligent control,” *Proc. IEEE*, vol. 77, no. 1, pp. 72–80, 1989.
- [13] M. Tavanpour, J. Mikhail, G. A. Wainer, G. Boudreau, H. Seyedmehdi, and R. Casselman, “File Transfer by Mobile User Collaboration,” Provisional Patent Filing

- Reference P46444 WO1. June 2015. PCT/IB2015/054524. USA.
- [14] M. Tavanpour, M. Moallemi, J. Mikhail, G. A. Wainer, G. Boudreau, and R. Casselman, "Shared Segmented Upload In Mobile Networks using Coordinated Multipoint, " Provisional Patent Filing Reference P43130 US1. February 2015. PCT/IB2015/051404. USA.
- [15] B. Cohen, "The BitTorrent Protocol Specification," 2008. [Online]. Available: [http://bittorrent.org/beps/bep\\_0003.html](http://bittorrent.org/beps/bep_0003.html). [Accessed: 15-Aug-2016].
- [16] M. Tavanpour, J. Mikhail, M. Moallemi, G. Wainer, G. Boudreau, and R. Casselman, "Data upload in LTE-A mobile networks by using shared segmented upload," *J. Networks*, vol. 10, no. 4, pp. 252–264, 2015.
- [17] M. Tavanpour, J. Mikhail, G. Wainer, M. Moallemi, G. Boudreau, and R. Casselman, "DEVS Based Modeling of Shared Segmented Upload in LTE-A Mobile Networks," in *Proceedings of the 18th Symposium on Communications & Networking*, 2015, pp. 60–67.
- [18] M. Tavanpour, M. Moallemi, G. Wainer, J. Mikhail, G. Boudreau, and R. Casselman, "Shared segmented upload in mobile networks using coordinated multipoint," in *Proceedings of the 2014 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS 2014 - Part of SummerSim 2014 Multiconference*, 2014, pp. 662–669.
- [19] R. W. Simons, "Guglielmo Marconi and early systems of wireless communication," *Gec Rev.*, vol. 11, no. 1, pp. 37–55, 1996.
- [20] G. Marconi, "Radio communications by means of very short electric waves," *IRE Trans. Antennas Propag.*, vol. 5, no. 1, pp. 90–99, 1957.
- [21] M. R. Bhalla and A. V. Bhalla, "Generations of mobile wireless technology: A survey," *Int. J. Comput. Appl.*, vol. 5, no. 4, 2010.
- [22] V. Zalud, "Wireless Cellular Mobile Communications," *Radioengineering*, 2002.
- [23] A. Ghosh, J. Zhang, J. G. Andrews, and R. Muhamed, *Fundamentals of LTE*. Pearson Education, 2010.
- [24] D. Seo, *Evolution and Standardization of Mobile Communications Technology*. IGI Global, 2013.
- [25] V. H. Mac Donald, "Advanced mobile phone service: The cellular concept," *Bell Syst. Tech. J.*, vol. 58, no. 1, pp. 15–41, 1979.
- [26] S. Kumar, *Wireless Communications Fundamental & Advanced Concepts: Design Planning and Applications*, vol. 41. River Publishers, 2015.
- [27] W. C. Y. Lee, "Elements of cellular mobile radio systems," *IEEE Trans. Veh. Technol.*, vol. 35, no. 2, pp. 48–56, 1986.
- [28] P. Nicopolitidis, A. S. Pomportsis, G. I. Papadimitriou, and M. S. Obaidat, "Wireless networks," 2003.
- [29] L. Goleniewski, *Telecommunications essentials: the complete global source for communications fundamentals, data networking and the Internet, and next-generation*

- networks*. Addison-Wesley Professional, 2002.
- [30] S. D. Bryen, *Technology Security and National Power: Winners and Losers*. Transaction Publishers, 2015.
  - [31] E. Dahlman, S. Parkvall, J. Skold, and P. Beming, *3G evolution: HSPA and LTE for mobile broadband*. Academic press, 2010.
  - [32] Y. Zaki, “Mobile Communication Systems,” in *Future Mobile Communications*, Springer, 2013, pp. 5–11.
  - [33] P. Chaudhury, W. Mohr, and S. Onoe, “The 3GPP proposal for IMT-2000,” *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 72–81, 1999.
  - [34] C. Cox, *An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications*. John Wiley & Sons, 2012.
  - [35] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.
  - [36] I. F. Akyildiz, D. M. Gutierrez-Estevez, and E. C. Reyes, “The evolution to 4G cellular systems: LTE-Advanced,” *Phys. Commun.*, vol. 3, no. 4, pp. 217–244, 2010.
  - [37] M. Sauter, *From GSM to LTE-advanced: An Introduction to Mobile Networks and Mobile Broadband*. John Wiley & Sons, 2014.
  - [38] B. T. Scheme, “LTE: the evolution of mobile broadband,” *IEEE Commun. Mag.*, vol. 45, 2009.
  - [39] M. Baker, S. Sesia, and I. Toufik, “LTE-The UMTS Long Term Evolution From Theory to Practice.” Chichester: John Wiley & Sons Ltd, 2011.
  - [40] Global mobile Suppliers Association, “GSA Evolution to LTE report: 480 LTE networks launched in 157 countries,” 2016. [Online]. Available: <http://gsacom.com/paper/gsa-evolution-to-lte-report-january-25-2016-480-lte-networks-launched-in-157-countries/>. [Accessed: 16-Aug-2016].
  - [41] S. Parkvall, E. Dahlman, A. Furuskar, Y. Jading, M. Olsson, S. Wänstedt, and K. C. Zangi, “LTE-Advanced-Evolving LTE towards IMT-Advanced,” in *VTC Fall*, 2008, pp. 1–5.
  - [42] J. Wannstrom, “LTE-Advanced,” 2013. [Online]. Available: <http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced>. [Accessed: 15-Aug-2016].
  - [43] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, “Coordinated multipoint: Concepts, performance, and field trial results,” *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102–111, 2011.
  - [44] J. Lee, Y. Kim, H. Lee, B. L. Ng, D. Mazzaresse, J. Liu, W. Xiao, and Y. Zhou, “Coordinated multipoint transmission and reception in LTE-advanced systems,” *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 44–50, 2012.
  - [45] E. Pateromichelakis, M. Shariat, A. ul Quddus, and R. Tafazolli, “On the evolution of multi-cell scheduling in 3GPP LTE/LTE-A,” *IEEE Commun. Surv. Tutorials*, vol. 15, no.

- 2, pp. 701–717, 2013.
- [46] S. Sun, Q. Gao, Y. Peng, Y. Wang, and L. Song, “Interference management through CoMP in 3GPP LTE-advanced networks,” *IEEE Wirel. Commun.*, vol. 20, no. 1, pp. 59–66, 2013.
- [47] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzaresse, S. Nagata, and K. Sayana, “Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges,” *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, 2012.
- [48] M. Jevtić, N. Zogović, and G. Dimić, “Evaluation of wireless sensor network simulators,” in *Proceedings of the 17th Telecommunications Forum (TELFOR 2009), Belgrade, Serbia, 2009*, pp. 1303–1306.
- [49] S. Siraj, A. Gupta, and R. Badgular, “Network simulation tools survey,” *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 1, no. 4, pp. 199–206, 2012.
- [50] S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers, “The state of peer-to-peer simulators and simulations,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 95–98, 2007.
- [51] M. Köksal, “A survey of network simulators supporting wireless networks,” *l{\'i}nea <http://www.ceng.metu.edu.tr/~e1595354/A%20Survey>*, vol. 20, 2008.
- [52] A. I. Concepcion and B. F. Zeigler, “DEVS formalism: A framework for hierarchical model development,” *IEEE Trans. Softw. Eng.*, vol. 14, no. 2, p. 228, 1988.
- [53] G. Wainer, “CD++: a toolkit to develop DEVS models,” *Softw. Pract. Exp.*, vol. 32, no. 13, pp. 1261–1306, 2002.
- [54] Q. Zhang and C. Yang, “Transmission mode selection for downlink coordinated multipoint systems,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 465–471, 2013.
- [55] A. H. Sakr, H. ElSawy, and E. Hossain, “Location-aware coordinated multipoint transmission in OFDMA networks,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 5166–5171.
- [56] A. Orsino, L. Militano, G. Araniti, A. Molinaro, and A. Iera, “Efficient data uploading supported by D2D communications in LTE-A systems,” in *European Wireless 2015; 21th European Wireless Conference; Proceedings of*, 2015, pp. 1–6.
- [57] L. Militano, A. Orsino, G. Araniti, A. Molinaro, and A. Iera, “A constrained coalition formation game for multihop D2D content uploading,” *IEEE Trans. Wirel. Commun.*, vol. 15, no. 3, pp. 2012–2024, 2016.
- [58] L. Militano, A. Orsino, G. Araniti, A. Molinaro, A. Iera, and L. Wang, “Efficient spectrum management exploiting D2D communication in 5G systems,” in *2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2015, pp. 1–5.
- [59] A. Al-Habashna, G. Wainer, G. Boudreau, and R. Casselman, “Cached and segmented video download for wireless video transmission,” in *Proceedings of the 49th Annual Simulation Symposium*, 2016, p. 15.
- [60] T. Chen, Y. Yang, H. Zhang, H. Kim, and K. Horneman, “Network energy saving

- technologies for green wireless access networks,” *IEEE Wirel. Commun.*, vol. 18, no. 5, pp. 30–38, 2011.
- [61] D. González, H. Yanikomeroglu, M. Garcia-Lozano, and S. R. Boqué, “A novel multiobjective framework for cell switch-off in dense cellular networks,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 2641–2647.
- [62] G. Nigam, P. Minero, and M. Haenggi, “Coordinated multipoint joint transmission in heterogeneous networks,” *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4134–4146, 2014.
- [63] J. Mikhail, “Model file for SSU model.” [Online]. Available: [http://www.sce.carleton.ca/faculty/wainer/wbgraf/lib/exe/fetch.php?media=model\\_sample:ssu\\_model\\_file.ma.zip](http://www.sce.carleton.ca/faculty/wainer/wbgraf/lib/exe/fetch.php?media=model_sample:ssu_model_file.ma.zip).
- [64] Third Generation Partnership Project, “3GPP Technical Report 36.942, V10.2.0,” 2011.
- [65] M. Hata, “Empirical formula for propagation loss in land mobile radio services,” *IEEE Trans. Veh. Technol.*, vol. 29, no. 3, pp. 317–325, 1980.
- [66] M. Alam, D. Yang, J. Rodriguez, and R. A. Abd-Alhameed, “Secure device-to-device communication in LTE-A,” *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 66–73, 2014.
- [67] Y. Huiyu, Z. Naizheng, Y. Yuyu, and P. Skov, “Performance evaluation of coordinated multipoint reception in CRAN under LTE-Advanced uplink,” in *Communications and Networking in China (CHINACOM), 2012 7th International ICST Conference on*, 2012, pp. 778–783.
- [68] N. Kong and L. B. Milstein, “Average SNR of a generalized diversity selection combining scheme,” *IEEE Commun. Lett.*, vol. 3, no. 3, pp. 57–59, 1999.
- [69] G. Cili, H. Yanikomeroglu, and F. R. Yu, “Cell switch off technique combined with coordinated multi-point (CoMP) transmission for energy efficiency in beyond-LTE cellular networks,” in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 5931–5935.
- [70] X. Zhang and X. Zhou, *LTE-advanced air interface technology*. CRC Press, 2012.
- [71] Third Generation Partnership Project, “3GPP Technical Report 36.843, V12.0.0,” 2014.
- [72] WINNER+, “D5.3: WINNER+ Final Channel Models,” 2010.
- [73] WINNER, “WINNER II Channel Models, D1.1.2, V1.2,” 2007.