Upload User Collaboration in the Data Upload for LTE-Advanced Networks

by

Misagh Tavanpour

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Carleton University

Ottawa, Ontario

To my Mother, Tahereh.

To my father, Mohammad.

To my love, Helia.

To the best siblings, Maryam, Mahdi, Milad.

Acknowledgements

I would first like to express my sincere gratitude to my supervisor Professor Gabriel Wainer for his brilliant guides and supports during my PhD program. He thought me a lot during these years.

My sincere thanks go to Dr. Gary Boudreau and Ronald Casselman from Ericsson Canada, for their insightful comments and great feedbacks. Without their precious support, it would not be possible to conduct this research.

My sincere thanks also go to my friend and colleague Jan Mikhail for his magnificent collaboration in the last couple of years. I would like to thank my former colleague Dr. Mohammad Moallemi for his kind supports as well. I thank to my colleagues from ARS-Lab for all of their supports during this program.

Finally, I would like to thank my thesis committee: Prof. Emil Petriu, Prof. Abdolreza Abhari, Prof. Ian Marsland and Prof. Sonia Chiasson, for their insightful comments and perceptive questions.

Abstract

There have been ongoing efforts focused on improving mobile network standards to support the increasing user demands of high data rate services. These efforts are more important for cell-edge users for whom a long distance to the cell's center, and the higher interference from neighboring cells, degrades their performance. Long Term Evolution Advanced (LTE-A), is a promising standard for the Fourth Generation of Mobile Systems (4G) mobile networks, and it uses a number of technologies to enhance users' performance regardless of their location in the coverage area. LTE-A employs a technique called Coordinated Multi-Point (CoMP) to provide high data rate services for cell-edge users. However, we need methods that are more advanced to improve the performance provided by the current standards. In this way, we can meet requirements of the International Mobile Telecommunication Advanced (IMT-Advanced), and can meet the requirements of future mobile systems, IMT 2020 (called the Fifth Generation of Mobile Systems - 5G -).

In this work, we present two novel methods for uploading large files from a User Equipment (UE) to multiple evolved Node Bs (eNBs), namely Shared Segmented Upload (SSU) and Upload User Collaboration (UUC). These methods aid cell-edge UEs with lower data rate to upload their data faster and assist the operators in providing services that are more consistent for their customers throughout the network's coverage area. In addition, we introduce an improved mechanism for retransmission of erroneous Transport Blocks (TBs) in the LTE-A mobile networks. We also propose a new concept called Super Set (SS) as a solution to the fixed coordination set problem. Finally, we present a method to extend the UUC to support handover for the UEs that need to change their host cell.

We used the Discrete Event System Specification (DEVS) formalism to model a distributed LTE-A mobile network using a non-cooperative algorithm, a CoMP algorithm, and the SSU and UUC algorithms. These DEVS simulations were used to evaluate the effectiveness of SSU and UUC under various scenarios, which included rural

and urban area settings. The simulation results showed that SSU and UUC could improve cell-edge users' upload performance and reduce the latency for a cell-edge UE to upload its data to the network. Moreover, we could see that UUC enhanced the non-cell-edge user upload performance as well.

Table of Contents

List of Tables

List of Figures

List of Acronyms

1G: First Generation of Mobile Systems

2G: Second Generation of Mobile Systems

3G: Third Generation of Mobile Systems

3GPP: Third Generation Partnership Project

4G: Fourth Generation of Mobile Systems

5G: Fifth Generation of Mobile Systems

8PSK: Eight Phase Shift Keying

ACK: Acknowledgement

AMPS: Advanced Mobile Phone System

AP: Access Point

ARQ: Automatic Repeat Request

AWGN: Additive White Gaussian Noise

BS: Base Station

BSC: Base Station Controller

BSS: Base Station Subsystem

BTS: Base Transceiver Station

CAGR: Compound Annual Growth Rate

CC: Chase Combining

CoMP: Coordinated Multi-Point

CQI: Channel Quality Information

CSI: Channel State Information

CU: Cellular User

D2D: Device-To-Device

DEVS: Discrete Event System Specification

DFD: Data File Descriptor

DFDC: Data File Descriptor Complement

DPS: Dynamic Piece Size

EDGE: Enhanced Data rates for GSM Evolution

EGPRS: Enhanced General Packet Radio Service

eICIC: enhanced Inter-Cell Interference Coordination

eNB: Evolved Node B

EPC: Evolved Packet Core

EPS: Evolved Packet System

EUTRAN: Evolved UMTS Terrestrial Access Network

FDMA: Frequency Division Multiple Access

FFSK: Fast Frequency Shift Keying

FM: Frequency Modulation

GMSK: Gaussian Minimum Shift Keying

GPRS: General Packet Radio Service

GSM: Global System for Communication Mobile

HARQ: Hybrid Automatic Repeat Request

HSDPA: High Speed Downlink Packet Data Access

HSPA: High Speed Packet Access

HSS: Home Subscriber Service

HSUPA: High Speed Uplink Packet Data Access

ICI: Inter-Cell Interference

ICIC: Inter-Cell Interference Coordination

IMT-2000: International Mobile Telecommunications-2000

IMT-Advanced: International Mobile Telecommunication-Advanced

IP: Internet Protocol

IR: Incremental Redundancy

ITU: International Telecommunication Union

LoS: Line of Sight

LTE: Long Term Evolution

LTE-A: Long Term Evolution Advanced

M&S: Modeling and Simulation

MAC: Medium Access Control

ME: Mobile Equipment

MIMO: Multiple Input Multiple Output

MME: Mobility Management Entity

MS: Mobile Station

MTX: Mobile Telephone Exchange

NAS: Non-Access Stratum

NLoS: Non Line of Sight

NMT: Nordic Mobile Telephony

NSS: Network and Switching Subsystems

OFDM: Orthogonal Frequency Division Multiplexing

OSS: Operation and Support Subsystems

PAPR: Peak to Average Power Ratio

PDCP: Packet Data Convergence Protocol

PDMA: Pattern Division Multiple Access

PDU: Protocol Data Unit

PGW: Packet Data Network Gateway

PHY: Physical

PMI: Pre-Coding Matrix Indicator

QoS: Quality of Service

QPSK: Quadrature Phase Shift Keying

RI: Rank Indicator

RLC: Radio Link Control

RNS: Radio Network Subsystems

RRC: Radio Resource Control

SAE: System Architecture Evolution

SCMA: Sparse Code Multiple Access

SDU: Service Data Unit

SGW: Serving Gateway

SISO: Single Input Single Output

SMS: Short Message Service

SRS: Sounding Reference Signals

SSU: Shared Segmented Upload

TACS: Total Access Communication System

TB: Transport Block

TDMA: Time Division Multiple Access

TTI: Transmission Time Interval

UE: User Equipment

UML: Unified Modelling Language

UMTS: Universal Mobile Telecommunications System

UPS: Uniform Piece Size

USIM: User Services Identity Module

UTRAN: UMTS Terrestrial Access Network

UUC: User Upload Collaboration

V&V: Verification and Validation

VoIP: Voice over IP

VPS: Variable Piece Size

WCDMA: Wideband Code Division Multiple Access

1 Introduction

Computer networks are one of the top engineering achievements of the 20th century. Since computer networks were invented, their use has increased steadily. One study shows that the number of the Internet user was about 16 million in 1995, which is about 0.4% of the world population at that year. This number reached to 3366 million by end of 2015, which is about 46.4% of the world population in the same year [1]. Another study found out that the World Wide Web required 4 years to attract 50 million users, while it took 13 years and 38 years for the television and the radio to reach this number of users respectively [2].

Computer networks can use two connection types: wired and wireless. In a wired network, cables connect the nodes of the network. Wireless networks nodes use radio waves to communicate with each other, and they transmit their signals over the air. The various advantages of wireless networks (reduced cost, mobility) attracted a large number of users. A study shows that there were about 380 million wireless subscribers in U.S. in 2015, and about 48% of the American householders use wireless services only [3].

A *mobile network* (also called *cellular*) is a kind of wireless network that uses radio signals to provide voice and data services for its subscribers that use a mobile device (normally a cell phone) called a *User Equipment* (UE). The network provides radio coverage over land areas called a *cell*. Each cell has at least one fixed transceiver. This transceiver has different names in different standards. It can be called *cell site*, *Base Station* (BS), *Node B* (NB) or *evolved Node B* (eNB). The first generation of mobile networks only provided voice communication services. Later advancements introduced new services, such as text messaging and data transmission/reception. The latter opened a way for introducing many other applications such as web browsing, online gaming and video streaming.

The first analog cellular system was introduced in 1979, and since then, they became very popular. A study published by Ericsson shows that in 2015, the global mobile subscriptions had reached 7.3 billion, which is almost equal to 99% of the world population in the same year [4][5]. Same study shows that the global mobile subscriptions will reach to 9 billion in 2021 [4]. In addition, mobile networks are changing the way in which the Internet users choose to access the Internet. A study shows that, mobile devices in 2009 and 2015 generated 0.7% and 33.04% of all the web page views traffic in order [6]. On the other hand, this means that the mobile users produce more data traffic. For example, a study shows that the monthly data traffic per a Smartphone would rise from 1.4 GB/month in 2015, to 8.9 GB/month in 2021 [4].

1.1 Research Motivations

Many mobile networks customers use these data intensive services, which require high data rates from their network providers. Service providers are now facing two problems: the large number of UEs that they need to service, as well as their high data rate demands. Moreover, the International Telecommunication Union (ITU) defined new sets of requirements for the newer generation of mobile systems, which are called IMT-Advanced (for what is marketed as 4G systems) and the International Mobile Telecommunication 2020 (IMT-2020). The service providers should meet the requirements introduced in the IMT-Advanced and IMT-2020 as well. For example, service providers should be able to provide suitable consistent services for their subscribers regardless of their location in the covered area. However, providing high data rates to the UEs in all the areas of coverage is challenging, especially when a UE is located near to a cell's border. This group of users has two problems: the distance to the cell's center (where their serving eNB is located) and the interference from the neighboring cells.

There are different ongoing efforts in order to deal with these problems. We can classify the current efforts into three categories: those that focus on improving the efficiency of the current resources, the design of new hardware, and the definition and provision of new standards. In terms of standards, service providers are seeking for new algorithms or

methods that are more efficient (compared to the existing ones) to improve both the network and user performance. One promising standard for the Fourth Generation of mobile networks, introduced by the 3rd Generation Partnership Project (3GPP), is called Long Term Evolution-Advanced (LTE-Advanced). Different forecasts show that the LTE network will have the highest Compound *Annual Growth Rate* (CAGR) among all other mobile standards between 2015 and 2021 [4]. The number of subscriptions in LTE networks is expected to increase from 500 million in 2015 to 4,300 million in 2021 [4].

One of the objectives of LTE-A networks (and the new standards in 5G as the next generation of mobile networks) is to provide consistent services for the UEs regardless of their location. To do so, mobile network standards like LTE-A use different techniques, one of which is called Coordinated Multi-Point (CoMP). CoMP uses a set of eNBs, called a coordination set, that work together to reduce interference and enhance the signal strength received. This form of the coordination is especially beneficial to the UEs located close to the cell's edge. Although CoMP improves the performance of cell-edge users, we still need methods that are more advanced to meet or exceed the IMT-Advanced and IMT-2020 requirements.

1.2 Research Goals

The primary goal of this research is to address the cell-edge UEs upload problems in a distributed CoMP architecture by proposing new algorithms for the cell-edge users, focusing in particular on uploads. The LTE-Advanced protocol was selected as the target communication protocol. The proposed solutions should work in the upper layers of the LTE-Advanced protocol stack. These solutions should also be able to work on the top of CoMP.

Another goal of this research is to evaluate the performance of the proposed methods. As discussed, mobile networks have attracted a large number of users, and almost every ten years a new generation of the mobile network standards were introduced, improving the systems. Among all of the different methods proposed for different standards, only a few are selected as candidate methods for a specific standard. Before using the new methods

in the mobile networks, they need to be tested them under various scenarios in order to check their validity and efficiency prior to implementation. Modeling and Simulation (M&S) is a popular method to address these issues. An efficient M&S technique reduces cost, saves time, provides desirable level of accuracy, and introduces a virtual environment that allows researchers to run their simulations in a safe and cost-effective manner.

In order to study the performance of the proposed methods, we need to use a proper M&S platform to model the mobile network components, the simplified LTE protocol stack, and the proposed algorithms. Moreover, we need to compare the proposed algorithms with existing standard methods to evaluate their performance. In particular, we need to model a standard non-cooperative method as well as a CoMP method. Finally, we need to study the performance of the proposed methods under the various simulation scenarios, including both urban and rural area settings.

1.3 Research Contributions

In this research, we present two advanced algorithms, called Shared Segmented Upload (SSU) and Upload User Collaboration (UUC), with the goal of uploading large files from the UEs to the mobile network in a distributed CoMP architecture. Besides these two methods, which have been patented during this Thesis [7][8], we defined a new method for the joint data recovery of the received Transport Blocks (TBs) at the different eNBs to increase the probability of this process and to reduce the number of the required retransmission by the UEs. This technique has also been patented [8]. Moreover, we discuss two other techniques, which were designed to improve the cell-edge users' performance by extending the UUC and expanding the coordination set concept. These methods have been patented as well [9].

1.3.1 Overview on SSU

SSU has a number of common points with the BitTorrent protocol [10]. The latter is used to speed up the download of large files on the Internet by allowing users to join a swarm of hosts to download and upload from each other simultaneously. SSU adapted this

technique to improve data upload from a UE to a set of eNBs. SSU can work over networks with lower bandwidth, and it can be considered a suitable upload method for cell-edge users in mobile networks. This technique is a subset of the Joint Processing method in CoMP. SSU uses small segments to transfer large files from a single UE to the group of eNBs that are coordinated dynamically. File segments are transmitted independently, and the eNBs with better communication channels with the UE can receive more segments. This process speeds up the data upload. Finally, the segments collected at different eNBs of the coordination set are gathered and organized at the serving eNB.

1.3.2 Overview on UUC

The UUC method is used for transferring files from a UE to the eNBs (or vice versa). In the upload process, each UE divides its data file into a number of pieces. The UE that holds the data file starts the upload process by transferring pieces to the group of eNBs that are coordinated dynamically. At a same time, this UE can ask for help from neighboring UEs, and let them upload some of the pieces. In fact, UUC focuses on enhancing the UEs upload process by using the upload power of multiple users that are close to each other. These UEs can communicate directly with each other using Device-To-Device (D2D) communications. Theoretically, these UEs can be either served by the same subset of eNBs or served by different subsets of eNBs. The idea is that each UE uses its communication channel to upload some of the pieces. In other words, different pieces of a file can be uploaded through multiple communication channels.

This method tries to speed up the upload process of the UEs regardless of their position in the cell (i.e., it can be used to enhance the upload in the cell's edge and with minimum variations it can be used outside the edge or in non-COMP scenarios). Likewise, this method can be employed to enhance the UEs download process.

1.3.3 Overview on the Super Set Concept

In the UUC [8], it is assumed that an owner UE and its helper UEs are served by same subset of the eNBs as their coordination set. However, this assumption (having a fixed

coordination set) is not necessarily optimal and by letting the UEs within the D2D cluster communicate with different subsets of eNBs as their CoMP coordination set can result in additional throughput gains. To do so, we define the Super Set concept that improves the UUC in such a way that the UEs that are involved in the upload process of an owner UE can use different coordination set of the eNBs. In other words, we want to let the owner UE and its helper UEs to use different eNBs set as their coordination set.

1.3.4 Overview on UUC Handover

In the UUC and the super set algorithms, we assumed that the owner UEs do not change their serving eNB or their host cell during the upload process. This assumption needs to be removed because in the real world the users can be mobile and possibly change their position in the network one or more times over the duration of the UUC upload. As a result, the operators should be able to address this kind of situation and provide consistent services for their user. In this work, we show how we can extend the UUC to support handover for the owner UEs that need to change their serving eNB in order to maintain a communications link of sufficient quality. In addition, we discuss the handover concept when we are dealing with super sets in the mobile network. To do so, we discuss the new steps and messages that we need to add to the LTE handover process to adapt it for the UUC algorithm.

1.3.5 Overview on the Improved Retransmission Mechanism

LTE-Advanced uses the Hybrid Automatic Repeat Request (Hybrid ARQ) method with soft combining as the retransmission policy for the UEs including the cell-edge UEs. This means that if none of the eNBs of the cell-edge UEs' coordination set can receive an error free TB, the serving eNB asks the cell-edge UE to retransmit the erroneous TB, and it uses the new received TB for data recovery in addition to the previous erroneous TB. The problem with this method is that each retransmission is an extra overhead for the cell-edge UE. In our proposed method for such a situation, the non-serving eNB sends a copy of the received TB to the serving eNB instead of the cell-edge UE retransmits the TB. After that, the serving eNB tries soft combining to recover original data from all the received copies of that TB. If the outcome of the soft combining process is not

successful, the UE retransmits that TB.

1.3.6 Related Publications

We published the outcomes of this research in three patents in a collaboration with Ericsson Canada. These patents cover the core ideas that we proposed, and we discuss all those ideas in this thesis in detail. Moreover, we published some of the obtained results in a number of articles in the different journals and conference proceedings. At the time of writing this thesis, we have two more articles (one journal paper and a conference paper) waiting for to be published. In the rest of this section, the list of related publication to this thesis is based on the date of publication in a descending order.

Misagh Tavanpour, Baha Uddin Kazi, Gabriel A. Wainer, "Discrete Event Systems Specifications Modeling and Simulation of Wireless Networking Applications", Submitted to SIMULATION, 2017.

Misagh Tavanpour, Jan Mikhail, Gabriel A. Wainer, Gary Boudreau, "Upload User Collaboration in Mobile Networks Using Coordinated Multipoint", Submitted to SpringSim 2017.

Misagh Tavanpour, Gary Boudreau, Gabriel A. Wainer, "File Transfer by Mobile User Collaboration: Super Sets and Handover", Provisional Patent Filing Reference P48914 WO1. PCT/IB2016/051382. March 2016, USA.

Misagh Tavanpour, "A Communication Protocol for Querying Information from a WIFI AP by a LTE Modem", Ericsson Internal Report, September 2015, Ottawa, Canada.

Misagh Tavanpour, Jan Mikhail, Gabriel A. Wainer, Gary Boudreau, Hossein SeyyedmMahdi, Ronald Casselman "File Transfer by Mobile User Collaboration", Provisional Patent Filing Reference P46444 WO1. PCT/IB2015/054524. June 2015, USA.

Misagh Tavanpour, Jan Mikhail, Mohammad Moallemi, Gabriel A. Wainer, Gary Boudreau, Ronald Casselman, "DEVS Based Modeling of Shared Segmented Upload in LTE-A Mobile Networks", Proceedings of 18th Communication and Networking Simulation Symposium (CNS'15), PP 60-76, April 2015, Alexandria, VA, USA.

Misagh Tavanpour, Mohammad Moallemi, Gabriel A. Wainer, Jan Mikhail, Gary Boudreau, Ronald Casselman, "Data Upload in LTE-A Mobile Networks by Using Shared Segmented Upload", Published in Journal of Networks, Vol 10, No 4, PP 252-264, April 2015.

Misagh Tavanpour, Mohammad Moallemi, Gabriel A. Wainer, Jan Mikhail, Gary Boudreau, Ronald Casselman, "Shared Segmented Upload in Mobile Networks using Coordinated Multipoint", Provisional Patent Filing Reference P43130 US1. PCT/IB2015/051404. February 2015, USA.

Misagh Tavanpour, Mohammad Moallemi, Gabriel A. Wainer, Jan Mikhail, Gary Boudreau, Ronald Casselman, "Shared segmented upload in mobile networks using coordinated multipoint", International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), PP 662-669, July 2014, Monterey, CA, USA.

Gabriel A. Wainer, **Misagh Tavanpour**, Emilie Broutin, "Application of the DEVS and Cell-DEVS formalisms for Modeling Networking Applications", Proceeding of Winter Simulation Conference (WSC13), PP 2923-2934, December 2013, Washington DC, USA.

Misagh Tavanpour, Gabriel Wainer, Gary Boudreau, Ronald Casselman, "DEVS-based Modeling of Coordinated Multipoint Techniques for LTE-Advanced", SIGSIM PADS Ph.D. Colloquium, May 2013, Montreal, Canada.

Misagh Tavanpour, Gabriel Wainer, Gary Boudreau, Ronald Casselman, "DEVS-based

Modeling of Coordinated Multipoint Techniques for LTE-Advanced", 16th Communications and Networking Symposium (CNS'13), April 2013, San Diego, USA.

1.3.7 Thesis Organization

The rest of this thesis is organized as follows: in chapter 2, the background of mobile networks and the different generations of the mobile systems are discussed. In addition, some of the related works that have been done in the same area as the topic of this thesis are reviewed briefly. The SSU and UUC algorithms, the Super Set concept, the UUC handover, and the proposed retransmission method are studied in detail in chapters 3, 4, 5, 6 and 7 respectively. The SSU DEVS models, simulation scenarios and the simulation results are presented in chapters 8. Chapter 9 presents these topics for the UUC algorithm. In chapter 10, we will conclude this research.

2 Background

The primitive mobile telephone systems used a powerful BS to provide radio coverage over a large geographical area. This type of BSs was able to provide radio coverage over an area between 40 to 60 miles. However, considering the available frequency band, this type of BS could support a limited number of communication channels simultaneously. Hence, a limited number of the customers could use the mobile telephone system at the same time. For example, the mobile telephone system in the entire metropolitan area of New York City could serve only 543 users at a time in 1976 [11][12].

The mobile devices that were being used in the mobile telephone systems were bulky and heavy. They were usually installed in vehicles. If a person wanted to use this service outside of his/her car, he/she required a briefcase for the mobile telephone. Moreover, those mobile devices consumed high power, and they required a considerable amount of time for recharging. Another important characteristic of the mobile telephone systems was the fact that they were regional systems, and mostly the governments or monopoly organization controlled them. Different companies launched their own mobile telephone systems in the different parts of the world, and these systems were not compatible with each other. This phenomenon became an issue when the mobile network operators decided to expand their networks, and offer newer services such as roaming to their users.

The mobile telephony systems operators realized that if they wanted to expand their systems, they needed to share their systems' specification among each other. Therefore, they introduced standards, which defined their systems specification. These standards assisted different vendors to produce their equipment based on the standardized platform, and helped the mobile technology to grow faster. The researchers categorized these standards based on their specifications into different family groups: First Generation of Mobile Systems (1G) to Fifth Generation of Mobile Systems (5G).

At the time of writing this thesis, the research on 5G networks has been started and the industrial experts expect 5G technologies being deployed by 2020. Moreover, the responsible international organization for the mobile network standardization process announced that the final specification of 5G technologies is going to be submitted in February 2020 [13].

2.1 First Generation of Mobile Systems (1G)

The first generation of mobile systems was introduced based on the analog transmission in the 1980s. These systems were based on the circuit-switched technology, and they were designed for the voice transmission. The use of cellular concept was the main difference between 1G and its predecessor (known as mobile telephone systems or 0G). Mobile telephone systems used a powerful BS, which transmitted a signal as far as it is possible. In the cellular concept the idea is to limit the range of the signal transmission over an area called cell, and handed off radio signals between the communication towers. In this way, the same frequency resources could be used in the different cells. This frequency reuse increases the system capacity compared to mobile telephony systems, and allows the support of more users at a same time [11][12]. Table 2.1 shows some of the characteristic of the 1G standard.

1G standard	NMT	AMPS	TACS
Year	1981	1983	1985
Region	Scandinavian	USA	UK
Downlink frequency band (MHz)	453 - 457.5	824 - 849	890 - 915
Uplink frequency band (MHz)	463 – 467.5	869 - 894	935 - 960
Modulation	FM	FM	FM
Duplex method	FDD	FDD	FDD
Multiple access method	FDMA	FDMA	FDMA
Channel spacing (KHz)	25	30	25
Transmission bit rate (Kbps)	1.2	10	8
Cell size (km)	2 – 30	2 – 20	2 - 20

Table 2.1: 1G standards technology

2.2 Second Generation of Mobile Systems (2G)

The 1G system had inconsistent voice quality with cross talk between users. They had unreliable handoff, and they could not provide acceptable security level for their users. In addition, those systems had limited capacity that could not support the growing rate of users. In the 1990s, the second generation of mobile communication systems (2G) was introduced based on the digital technology to provide better services for more customers. The digital transmission can be considered as the most important difference between 1G and 2G. Moreover, 2G systems had three main benefits over their predecessors including encrypted phone conversations, efficient usage of the available spectrum, providing data services [14]. In case of the latter, 2G systems, introduced Short Message Service (SMS) as the first data service in the mobile networks. Later on, newer facilities were added to the 2G systems that enabled them to provide more data services like web browsing for their customers.

The Global System for Communication Mobile (GSM) is a packet switch protocol that is a major enhancement in the 2G system. Before GSM, different European countries used different technologies and protocols for mobile communication. This approach had some problems. There was no guaranty that a user device, which could work in one of these mobile networks, be able to work in another one.

The European countries agreed on developing and deploying a common cellular system to support the requirements of their users in any time or location across Europe. It was employed in non-European countries as well. Some of the key design goals of GSM include offering good speech quality, supporting international roaming, offering proper spectral efficiency, and offering low cost [15]. GSM is a circuit-switched system that was designed as a 2G mobile network. It uses Gaussian Minimum Shift Keying (GMSK) as the modulation method. In addition, GSM employs both Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) as the access technology to share the available bandwidth among the users. GSM-900 (the number presents the frequency band) uses 890-915 MHz band for the uplink and 935-960 MHz for the downlink. Moreover, in the GSM system, the channel spacing is 200 KHz and the cell

radius can be up to 35 km.

The rapid growing rate of Internet made it clear to the mobile network operators that a successful mobile network should be able to support Internet based services and applications. To do so, the core network had to be improved to able to support the Internet Protocol (IP). In addition, higher data rate was required for services like web browsing and video conferencing. As a solution to these issues, General Packet Radio Service (GPRS) was introduced in GSM by adding packet switching protocols in to GSM during the second half of the 1990. The 2G systems that were using packet switch domain in addition to circuit switch domain are known as the 2.5G system [16]. GPRS allows the operators to run the Internet Protocol (IP) standard over the core network for integrated voice and data application [17].

Another evolution to the GSM systems is known as Enhanced Data rates for GSM Evolution (EDGE) or Enhanced GPRS (EGPRS). 2.75G is the term usually used to refer to EDGE systems. The EDGE system provides higher data rate than GPRS. EDGE systems use Eight Phase Shift Keying (8PSK) as the modulation technique. Despite all the required changes, the cost of the upgrade of the GSM system to the EDGE system is relatively small [18]. By December 2010, the GSM family (GSM, GPRS, and EDGE) was launched in more than 200 countries across the world and it had more than 79% mobile networks market. The GSM family had more than 4 billion subscribers, and it was the dominant mobile network standard at the end of 2010 [19].

2.3 Third Generation of Mobile Systems (3G)

The International Telecommunication Union (ITU) is one of the agencies of the United Nations (UN). It is responsible for the related issues to the information and communication technologies [20]. In the middle of 1980s, it introduced International Mobile Telecommunications - 2000 (IMT-2000) to create a global standard for the wireless networks. The main characteristics of the IMT-2000 systems include worldwide usage, application in all radio environments (LAN, cellular, satellite, etc.), supporting wide range of services (voice, data, internet, etc.), and providing high data rates up to 2

Mbps [21]. Third generation (3G) of mobile systems is a set of standards and protocols that meets the requirements of IMT-2000.

There were parallel investigations on the 3G systems in the different part of the world in the 1990s. These parallel activities were problem to the globalization goal, because they were extending different developments. In December 1998, a number of standard development organizations from different regions of the world formed the Third Generation Partnership Project (3GPP) to develop a unified technical specification and technical reports for a 3G mobile systems [22].

3G coverage is extending quickly, and in 2015, 3G systems provided coverage for 69% of the world population [23]. There are different types of 3G networks such as Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), and Long Term Evolution (LTE) with different types of the core network. UMTS is based on the evolved GSM core network, and it was expected to provide worldwide access and global roaming for a wide range of services. Moreover, new specifications have been developed within the framework of 3GPP as the 3G evolution to improve the UMTS system. These technologies are known as 3.5G or 3.75G. HSPA and LTE are two samples of the evolved 3G technology.

UMTS architecture can be divided into two main domains: namely the User Equipment (UE) and the Infrastructure [24]. The Infrastructure is further divided into the Access Network and the Core Network. The Core Network includes the physical entities to provide support telecommunication. The Core Network includes a Serving Network responsible for routing calls and data, the Home Network is responsible for the management of subscription information and user specific data and the Transit Network is employed when the remote party is located outside of the network of the originating UE [24][25]. UMTS was developed based on the GSM architecture that also included GPRS and EDGE (Figure 2.1). The countries that they were using GSM networks agreed on using a new frequency band for UMTS networks. This new frequency band provided new capacities for the mobile networks operators. In addition, this new technology

required new radio access network as well, and this increases the cost of switching from 2G mobile systems into this 3G system. UMTS uses Quadrature Phase Shift Keying (QPSK) as the modulation technique, and it is not backward compatible with GSM systems. It was developed based on Wideband Code Division Multiple Access (WCDMA) as the radio access technology [26].

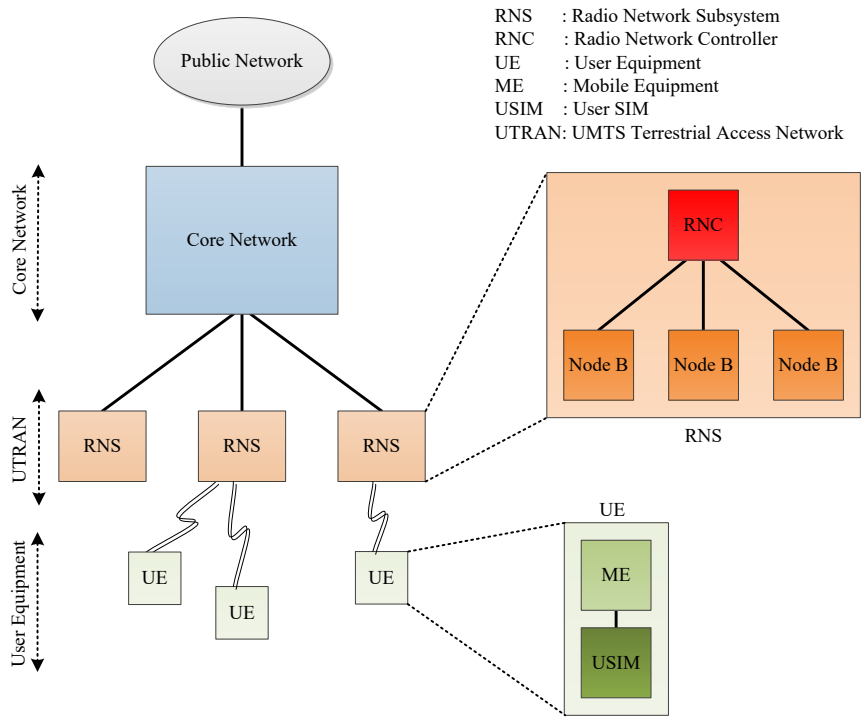


Figure 2.1: UMTS architecture

The High-Speed Packet Access (HSPA) standard is an evolved version of the UMTS standard. HSAP was introduced to improve the UMTS systems data speed and latency. It increased the download data rate up to 14Mbps, and it enhanced the upload speed up to 5.7Mbps. HSPA uses the available spectrum more efficiently to provide higher speed for the different applications like Video Telephony, File Upload, and Voice over IP (VoIP). In addition, HSPA has a lower latency compared to the UMTS system, which makes it a better choice for the real time applications. The HSPA architecture is similar to the UMTS architecture plus some hardware and software changes (mostly software changes) [27].

The second UMTS evolution in the 3GPP platform focused on the two work items including the radio network evolution and the core network evolution. LTE focused on the radio network evolution, and the result is called Evolved UTRAN (E-UTRAN). System Architecture Evolution (SAE) focused on the core network evolution and the result is called Evolved Packet Core (EPC). The EPC is a packet-only switch network. It is responsible for the non-radio related functionality of the mobile network (authentication, charging, etc.) as well as providing access to the external networks (Internet). EPC has a flat architecture. It is composed of a number of logical components: the Mobility Management Entity (MME), Serving Gateway (S-GW), the Packet Data Network Gateway (P-GW), and Home Subscriber Service (HSS). The E-UTRAN and the EPC together are referred as the Evolved Packet System (EPS). Figure 2.2 represents the EPS architecture.

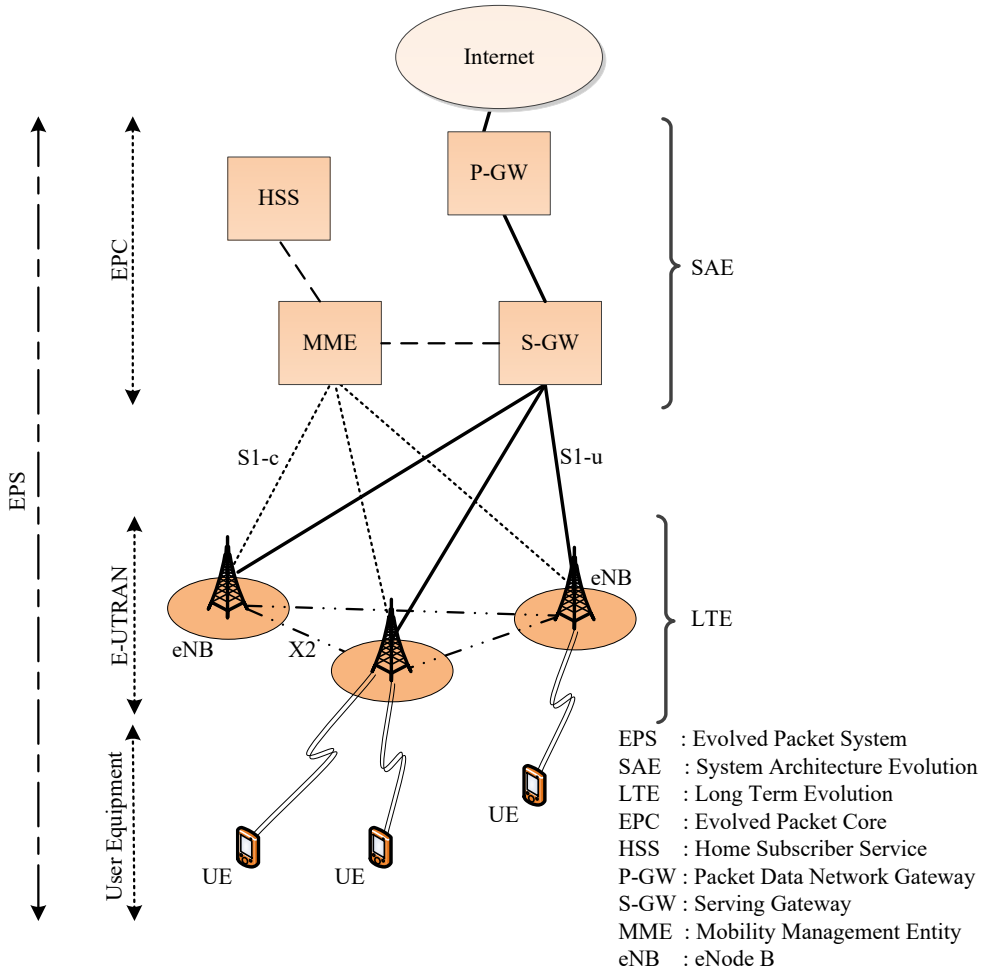


Figure 2.2: EPS architecture

MME is the control-plane node. It deals with the connection/release of bearers to/from a UE, tracking and paging of an UE in the idle mode, etc. S-GW is the user-plane node, which delivers the IP traffic between the UE and the external network. P-GW connects EPC to the external networks, and it is responsible for the allocation of IP addresses to UEs, policy control, etc. HSS is a database that contains the subscriber information [28]. E-UTRAN (or the LTE radio access network) is responsible for all the radio related functionality of the network such as the scheduling and the retransmission. Same as EPC, E-UTRAN has a flat architecture with a single type of logical node, which is called Evolved Node B (eNB). The latter is the evolution of Node B from UMTS. There are two identified interfaces in the E-UTRAN standard: S1 and X2. The eNBs use S1 to be connected to MME and S-GW. X2 connects eNBs to each other.

The work on LTE started in 2004, and the first LTE specification was introduced in the release 8 of 3GPP in 2008. The first commercial LTE network was launched in 2009. The LTE standard is part of the 3GPP 3G family standard (known as 3.75G). Release 8 was followed by other 3GPP releases such as release 9, 10 and 11. The latter releases introduced new capabilities and functionalities into the LTE standard. Figure 2.3 shows the overall E-UTRAN protocol architecture. This figure represents the LTE protocol architecture from both the user plane and the control plane prospective [29].

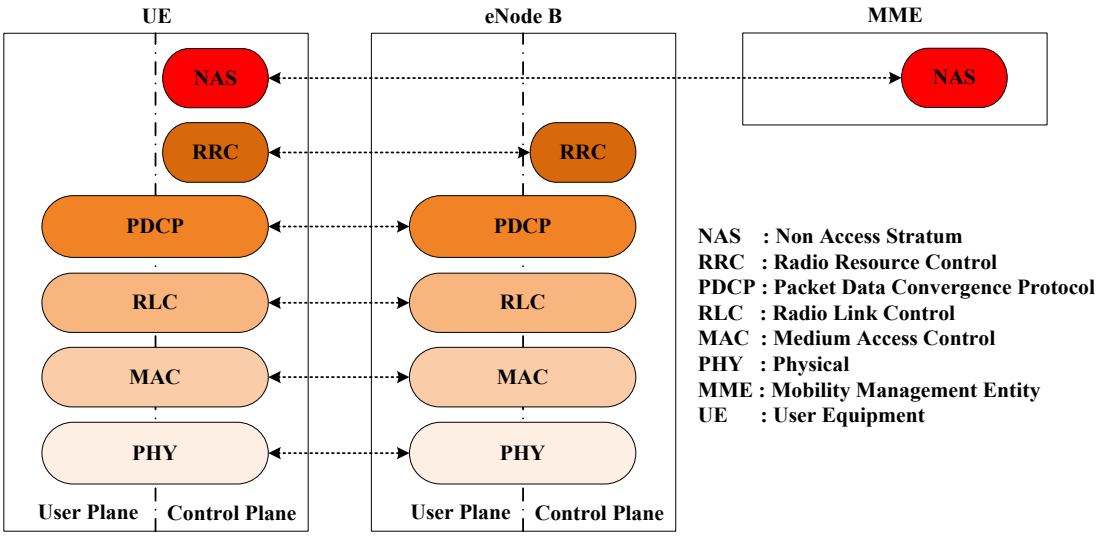


Figure 2.3: E-UTRAN protocol architecture

LTE UEs require establishing a signaling connection when they access the mobile network. This signaling occurs between the UE and MME. The Non Access Stratum (NAS) layer is responsible to handle this process. This layer is responsible for the mobility management and the session management of UEs. The Radio Resource Control (RRC) layer is responsible for the signaling between UEs and eNBs. The NAS messages are transported via the RRC layer, either encapsulated in dedicated RRS messages or concatenated with other RRC messages. Some of the other duties of RRC layer include broadcasting of the system information, radio bearers' management, UE management report and control of the reporting, Mobility functions, etc. The Packet Data Convergence Protocol (PDCP) layer provides services to the RRC and NAS layers. It is responsible for IP header compression, ciphering, in sequence delivery, etc. The Radio Link Control (RLC) layer handles for segmentation, concatenation, retransmission handling, duplicate detection and in sequence delivery to the PDCP layer. Also in case of retransmission, Automatic Repeat Request (ARQ) is performed by this protocol [29][30][31][32].

The Medium Access Control (MAC) layer is responsible for the link adaptation, mapping between the logical channels and the transport channels, error correction by mean of the Hybrid ARQ (HARQ) protocol. Likewise, this layer is responsible for the scheduling, and the scheduler is part of MAC layer. The main operation of the scheduler is called dynamic scheduling. It means that each one millisecond, the scheduler of the eNB takes the scheduling decisions, and the eNB sends this information to the selected UEs. In addition, the scheduler informs the RLC layer about the amount of data that can be sent in the next time interval. Therefore, the RLC layer can make a decision whether to perform segmentation or concatenation [29][32].

The Downlink scheduling and the uplink scheduling are separated in the LTE standard, and they are taken based on the downlink and uplink channels status respectively. In the case of downlink scheduling, the UE provides Channel State Information (CSI) report for the eNB to inform it about the instantaneous downlink channel status. CSI includes several different types of indicator including Channel Quality Information (CQI), Pre-Coding Matrix Indicator (PMI) and Rank Indicator (RI). CQI represents the quality of the

channel between the UE and eNB. The UE uses PMI and RI to suggest to the eNB about the pre-coder matrix that it can use for transmission to the UE. However, the eNB can ignore the UE recommendation and choose another pre-coder matrix. In this situation, the eNB should inform the UE about the new pre-coder matrix. In case of uplink scheduling, the UE sends Sounding Reference Signals (SRS) on the uplink to allow the eNB to estimate the uplink channel quality. This process can occur in two different ways: periodic SRS transmission and aperiodic SRS transmission [29][32][33].

The Physical (PHY) layer performs the actual transmission over the radio interface. It is responsible for coding/decoding, modulation/demodulation, etc. It provides services to the MAC layer in form of transport channels [29][32].

LTE design targets are discussed in [34]. Compared to the previous 3GPP technologies, LTE defines higher data rates as its target peak data rates for the downlink and the uplink (100Mbps and 50Mbps in order). Considering the 20 MHz spectrum, this is equivalent to a peak spectral efficiency of 5 bit/s/Hz in the downlink and 2.5 bit/s/Hz in the uplink. LTE supports both the FDD operation and the TDD operation. In case of FDD, the LTE specifications allow for the simultaneous uplink and downlink transmission at the peak data rates. In case of TDD, the reception and the transmission cannot happen at a same time, therefore the peak data rate requirement cannot be met simultaneously. An LTE system should be able to support at least 200 active UEs per cell for the spectrum allocation up to 5MHz. For the higher spectrums, this number is 400. For inactive UEs these numbers are not explicitly stated, but it is expected to be much higher than this. In addition, LTE should be able to support different coverage scenarios, for which the performance targets can be met. For the cells up to 5 km cell range, the user throughput, the spectrum efficiency, and the mobility requirements should be supported in the highest level. For the cells up to 30 km cell range, a slight degradation in both of the user throughput and the spectrum efficiency is tolerable.

To meet these designs targets, LTE uses number methods defined in release 8 and 9 of the 3GPP releases. Some of the basic principles behind the LTE are as following [35]:

- ✓ Orthogonal Frequency Division Multiplexing (OFDM) transmission:

This is a kind of multi-carrier transmission, which uses large number of narrowband subcarriers (straightforward multi-carrier transmission in HSPA uses few subcarriers with relatively wide bandwidth). The number of OFDM subcarrier depends on the subcarrier spacing. The basic subcarrier spacing for 3GPP LTE is 15 KHz. Also, it worth to mention that based on the LTE standard, either of QPSK (2 bit per symbol) or 16QAM (4 bit per symbol) or 64QAM (6 bit per symbol) can be used as the modulation technique

- ✓ Channel-dependent Scheduling:

Considering the dynamic nature of the resource requirement of the packet-data communication, shared-channel transmission is a good option to share the time-frequency resources among the users dynamically. In this manner, the scheduler determines which part of the shared resources should be allocated to which one of the user each time.

- ✓ Hybrid ARQ:

It allows the UE to send a request for the retransmission of an erroneously received transport block rapidly.

- ✓ Inter Cell Interference Coordination (ICIC):

In the LTE systems, frequency reuse coefficient is one. Therefore, same carrier frequencies can be used by the neighboring cells, and this can increase the interference between the neighbor cells and reduces the signal to interference ratio. Thus, ICIC recommends that the neighbor cells and their eNBs work in a coordinated manner to avoid sever interference. This can be done by exchanging set of predefined messages among the eNBs via X2 interfaces.

- ✓ Multi-antenna transmission:

This term refers to set of techniques that they try to enhance the signal to noise ratio by use of multiple antennas at the transmitter and/or receiver. These techniques include receive diversity, transmit diversity, beam forming and spatial multiplexing. In the latter, multiple antennas are used at both the transmitter side and the receiver side (the term

Multiple Input Multiple Output (MIMO) antenna processing is often used for this technique as well). Use of multi-antennas transmission is a key in the LTE systems to meet the aggressive LTE performance targets.

- ✓ Multicast and broadcast support:

The multi-cell broadcast requires the transmission of identical signals (which carry same information) from multiple synchronized cells. This method enhances the signal strength and eliminates the inter-cell interference.

The LTE networks deployments experience a meaningful growth. Between 2009 and 2014, there were 500 million LTE accumulated mobile subscriptions. The predictions show that this number reaches to 3100 million in the next five years (2015-2020) [36]. In 2010, there were 11 countries, in which LTE networks were lunched. By end of 2015, there were LTE networks in 151 countries[37] .

2.4 Fourth Generation of Mobile Systems (4G)

The future mobile networks will need to support a large numbers of UE with high data rate demands. Providing uniform high quality services over the covered geographical areas is a difficult task. To fulfill this goal, service providers constantly investigate on the new protocols to improve the quality of service for the UEs. In addition, to support users even more, ITU defined a new set of requirements for the newer generation of mobile systems, which is called IMT-Advanced (for what is marketed as 4G systems). Support of at least 40 MHz bandwidth, peak spectral efficiencies of 15 bit/s/Hz in the downlinks, peak spectral efficiencies of 6.5 bit/s/Hz in the uplinks and reduced control and user plane latency are some of those requirements for the IMT advances systems.

A candidate for the 4G systems is Worldwide Interoperability for Microwave Access (WiMAX). It is based on IEEE 802.16 standard, and it is designed in such a way to deliver broadband access services to everywhere at any time [38]. On the side, 3GPP also defined new technologies to meet or even exceed the IMT advanced requirements. Releases 10 and 11of 3GPP define new enchantments in the LTE protocol to fulfill the

IMT advanced requirements. The improved LTE systems (based on Releases 10 and 11) are often referred as LTE Advanced systems.

2.4.1 LTE-Advanced

LTE-Advanced has been standardized by 3GPP as a backward-compatible enhancement of LTE [39]. This standard meets or exceeds the IMT-Advanced requirements and is considered as a candidate for IMT-Advanced systems [40][41]. To overcome the transmission barriers such as Inter-Cell Interference (ICI) and to support high data rates as well as meet the IMT-Advanced requirements a number of technologies including advanced MIMO, wireless relays, heterogeneous deployments, enhanced Inter-Cell Interference Coordination (eICIC) and Coordinated Multipoint (CoMP) are employed in LTE-Advanced [42].

Advanced MIMO refers to expanding the downlink or the uplink spatial multiplexing to support more transmission layers. Wireless relay is a low-power eNB that wirelessly is connected to a part of the network in order to provide coverage for users on those areas (Figure 2.4).

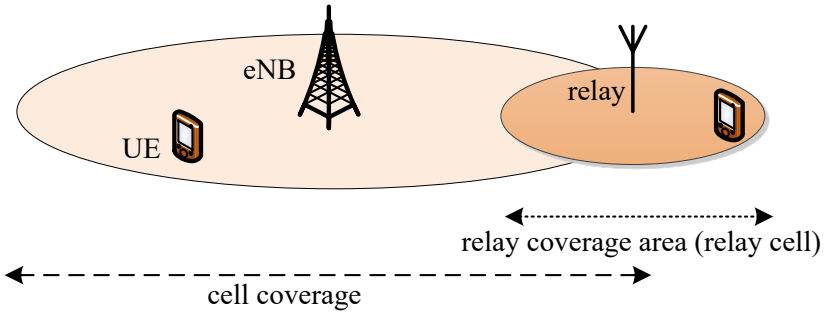


Figure 2.4: Relay cell

Heterogeneous deployments refer to the mixture of high-power macro nodes (eNBs) and the low-power ones (such as Pico nodes, Femto nodes and relays) with overlapping geographical coverage. LTE advanced use this technique to bring the network closer to the users in order to provide better services (Figure 2.5).

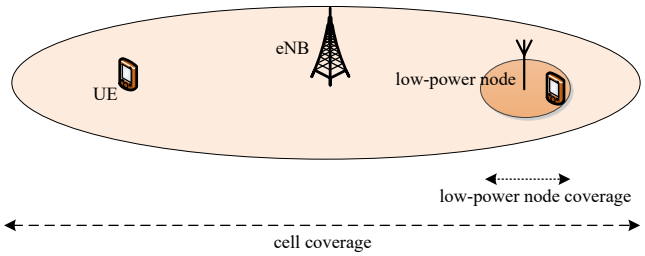


Figure 2.5: Heterogeneous Network (HetNets)

One of the objectives of LTE-A networks is to provide consistent services for the UEs regardless of their location. However, providing high quality signals to UEs in all coverage areas is challenging, especially when a UE is located near a cell border. This group of users has two problems: the long distance from the cell center where their serving eNB is located and the higher interference from the neighboring cells (*Inter-Cell Interference* (ICI)). The latter is a major bottleneck for the cellular networks performance [43]. In particular, this problem affects cell edge users' performance. It also acts as a barrier for the mobile network standards coming close to their theoretical rates [44]. ICI is a result of using the same radio resources in different cells in an uncoordinated way. To overcome these problems different types of techniques such as interference cancellation and interference coordination have been investigated [41][45][46][47]. Service providers require that these problems be addressed to meet the expectations of cell-edge users. CoMP is considered as the key solution of LTE-Advanced standard to the cell-edge users' problem. CoMP coordinates the eNBs to decrease the interference and increase the received signal power. ICIC methods are suitable for semi-static coordination among the eNBs, while CoMP technique covers coordination that is more dynamic [48].

CoMP refers to a set of eNBs that are coordinated jointly and dynamically. With the implementation of CoMP, eNBs can support joint scheduling of transmissions, and provide joint processing of the received signals. In a CoMP scenario, eNBs form coordination sets for which the main objective is to manage the interference to enhance the performance of UEs, especially for the cell edge users [49]. By coordinating and combining signals from multiple antennas and eNBs, it is possible for mobile users to have high quality and consistent performance when they require high-bandwidth services

for different applications. To support this feature in LTE-Advanced networks, eNBs and UEs require the exchange of scheduling decisions, hybrid ARQ feedback, channel state information (CSI) and other control information with each other [41]. In addition, the eNBs share the received messages from their UEs with other eNBs in coordination set through the 3GPP standard interface denoted as X2 [41]. This method has some drawbacks as well. Compared to a non-cooperative method, CoMP requires a backhaul with higher capacity and lower latency. In addition, it imposes overhead on the backhaul as well as more complexity to the mobile system.

Considering the way that control information is made available at the different transmission points, CoMP can be implemented in two ways: centralized and distributed [50]. In the centralized CoMP transmission approach, a central unit is the entity where all channel information and data from all UEs in the supported area by coordination set are available. This central entity can be an assigned eNode B or a higher order entity in the LTE network. For downlink transmissions UEs estimate the channel status and then they feedback this information to the serving cell. Once the serving cell receives this information from its UEs, it forwards this information to the central unit that is responsible for the scheduling operations. After computing the parameters related to the scheduling, the central unit sends the results to the coordinated eNBs in the coordination set. The main challenge in this architecture is the latency parameter to support effective exchange of information between eNBs in the coordination set. In addition, because all eNBs will need to send all of the UEs status information and data to the central unit there will be significant signaling overhead on the backhaul [39][40][19][51].

In distributed CoMP, the UEs send back the channel status to their serving eNBs in the coordination set and this information will be forwarded from the serving eNBs to the coordinating eNB. Hence, each eNB receives all of the UE feedback, including that related to other eNBs in the coordination set, and each eNB can perform its scheduling operation in a coordinated manner. It is worth mentioning that the schedulers are identical hence similar inputs result in similar outputs. The main advantages of this architecture are reduced infrastructure cost and signaling protocol complexity. These

benefits are possible because there is no dedicated central unit in this architecture, which results no need for eNBs to communicate with it, and hence, there is no need for communication links between a central entity and the CoMP eNBs. It should be noted that in a distributed architecture an eNB might be selected as a temporary CoMP coordination entity for a given CoMP session. A serious problem in this kind of architecture is handling the errors on the same feedback information on the different feedback links [39][40][19][51]. Figure 2.6 shows an example of distributed CoMP architecture.

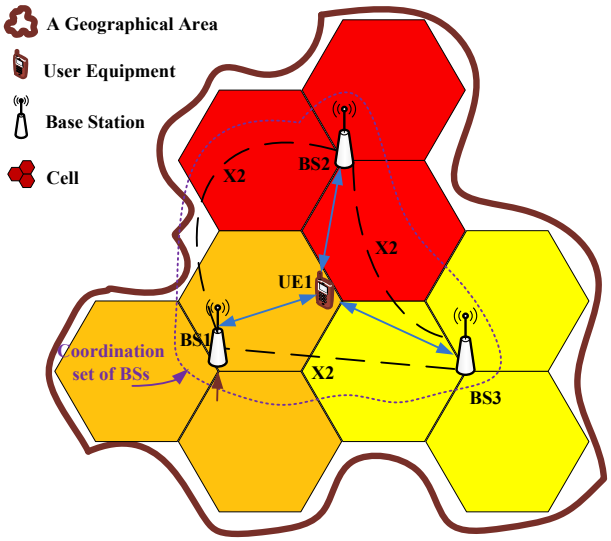


Figure 2.6: Coordination set in distributed CoMP transmission approach

There are two schemas for CoMP in LTE-Advanced with respect to the way the data and scheduling information is shared among eNBs: Coordinated scheduling/Beamforming and Joint Processing. In the latter approach, the eNBs in the coordination set share their data as well as the channel state and scheduling information with other eNBs. In the former approach, the exchange of data is not required and the eNBs just need to share the channel state information and the scheduling information. In other words in the Joint Processing scheme (Figure 2.7) the data to be transmitted to a single UE, is transmitted from eNBs simultaneously in coordination set. This increases the signal quality at the UE side and decreases the interference level. However, at same time the amount of data that needs to be exchanged over the backhaul is very large. In Coordinated

Scheduling/Beamforming, one of the eNBs in the coordination set (the serving eNB) serves each UE and the scheduling decisions are selected in a way to control interference among the eNBs in the coordination set. Therefore, in this case the eNBs just need to share scheduling information and the UE data does not need to be conveyed to all eNBs in coordination set since there is only one serving eNBs for one particular UE at any given scheduling instance [39][41][42][48][19].

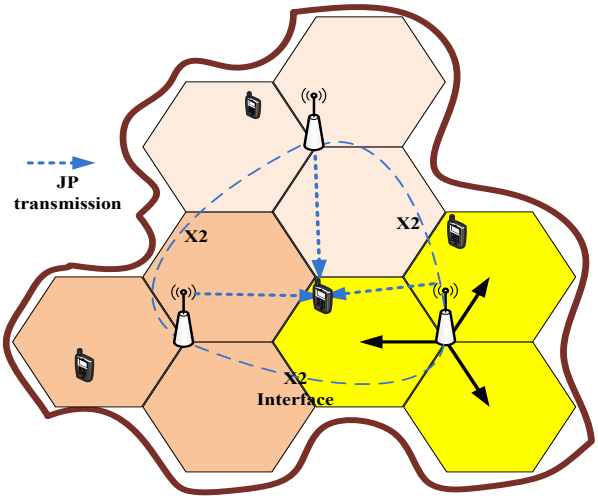


Figure 2.7: Joint Processing transmission in LTE-Advanced

2.5 Related Work

In [52], the authors revealed some of the reasons of the rapid increase of the traffic load in the mobile networks including the evolution of smart phones. They claimed that the current mobile networks are not flexible enough to overcome to the ever-increasing demands of the users of the mobile networks, and they will reach to their limit. They introduced HetNets as a cost-effective solution into this problem. They mentioned that the traditional mobile network infrastructure requires to be upgraded to a multi-tier network with smaller cells such as Micro, Pico and Femto cells overlaying the traditional macro cell. In result, both the capacity and coverage of the mobile networks will be improved. On the other hand, the authors also pointed out that the usage of the D2D communication would offer a range of opportunities to improve the spectrum efficiency of the mobile networks. However using these two techniques together raises another

serious challenge, which is the interference management between the different tiers within the heterogeneous architecture as well as between the infrastructure-to-device and D2D communication. Therefore, to benefit from opportunities offered by HetNets and D2D, an intelligence resource-allocation technique is required. In [52], the authors proposed an interference-aware resource-scheduling algorithm to address this problem.

The authors of another article ([53]) also mentioned that due to evolution of wireless networks, the traditional mobile network infrastructure is expected to be replaced by an heterogeneous architecture. They discussed that mobile data traffic grows rapidly, and wireless networks are more prone to face the bottleneck compared to any other part of the core network. On the other hand, they mentioned the successful experience of employing Content Delivery Networks (CDNs), where content cashing means that the content of servers is replicated at locations closer to the users. This approach offers reduction on the usage of network resources. Therefore, in this article, the authors claimed that the content cashing concept can be used to enhance the performance of mobile networks and merging content cashing concept within the wireless network is a promising paradigm. Moreover, limited battery of hand-held devices was mentioned as one of the important issues of wireless networks. To deal with this issue, they proposed a method to reduce energy budget of UEs in the upload process.

In [54], the authors mentioned that the use of the D2D links can improve the mobile network performance in some scenarios. In particulars, whenever the UEs are close enough to each other (the communication is in a local scope), the D2D links offer more efficient performance compared to the conventional communication via the eNBs. The authors of this article discussed that the D2D links also improve the mobile networks in other aspects such as extending the cell coverage, offloading cellular traffic and supporting content sharing in a neighborhood. In this paper several scenarios, in which the D2D communication improves the uplink performance, were discussed. The authors of this paper tried to improve the data upload in a single LTE-Advanced cell by designing and analyzing of D2D-based techniques. Using the proposed method in [54], a UE with a weak communication channel with its serving eNB, can forward its data to a close UE

that holds a high-quality channel with the serving eNB. After that, the latter UE uploads the data in behalf of the former UE. Moreover, the authors proposed a D2D-based schema for a cooperative use of the radio resources assigned to the UEs that use D2D communications.

Same group of authors in [55] discussed that 5G systems would use the D2D communications as one of the key technologies. They used a similar concept (the one in [54]) to propose a method to decrease the amount of radio resources required to upload a certain amount of data to an eNB. Same as previous article, the UE with a poor link uses D2D channel to send its data to another UE with a high-quality uplink channel, and the latter UE uploads the data as its own data. They also showed that compared to a standard LTE uploading schema, the proposed method reduces the energy consumption of the UEs for uploading same amount of data.

In [56], the authors extend the previous D2D-based schema, and they use multi hop D2D communication for data uploading toward the eNB to improve the user performance. This means that, a UE can use the D2D communication to send its data to a second UE, and the second UE uses the D2D communication to forward this data to a third UE that has better uplink communication channel. Finally, the third UE transfer the data of the first UE to the eNB. In other words, in this technique, a set of UEs cooperate to upload their data to an eNB by forming a multi hop D2D chain. In this method, only the UE at the head of the chain is responsible to upload the received contents from other UEs of the chain to the serving eNB. The head UE is the one with the best channel quality with the eNB. The authors also proposed that the UE at the head of chain could receive all the upload resources that the eNB wants to assign to the different UEs of the chain.

In [57], the authors used the D2D communication concept to propose a mechanism to enhance the coordinated multipoint transmission for the uplink between a UE and its serving eNB. To do so, the idle UEs are employed as the cooperating UEs. According to this proposal, the cooperating UEs receive the uplink signal from the primary UE and retransmit this signal to the serving eNB. The authors also extend this method to enhance

the users downlink as well. In case of the downlink communications, the cooperating UEs receive the downlink transmission from the serving eNB and forward the received data signal to the primary UE.

In another study [58], the authors mentioned the important role of D2D communication in the future mobile networks. They also discussed that an optimized uplink resource sharing between the cellular users (CUs) and D2D links is necessary for the proper performance of the D2D links. They allowed multiple D2D links share the resources of each CU, and each D2D link reuses the resources of several CUs. Then they proposed an algorithm to optimize the transmit power of the D2D links and the CUs. The authors in [59] discussed same concern. They mentioned that the D2D communications will be an essential part of the 5G systems, and they will be useful to reduce the overload of the core network. However, same as the previous article, they claimed that an efficient resource-sharing schema between D2D pairs and CUs is a serious issue that needs to be addressed. They discussed that this problem requires a dynamic solution and letting D2D pairs to use the resources of a specific CU for a long time may not an efficient approach. Finally, the authors proposed a D2D scheduler that allocate resources and set the transmission power in a dynamic manner.

Carrier aggregation is a way to increase the UEs' bandwidth in order to provide users high data rate services. However, there are some challenges like intense increase in the peak-to-average-power-ratio (PAPR) that required to be addressed before we can use CA techniques. In [60], the authors proposed a method to reduce PAPR for the UEs' uplink in the LTE-Advanced mobile networks. Their method is based on two stages of frequency and time domain processing in order.

Multiple antennas systems have attracted lots of attention in the current trend of the mobile networks. MIMO systems provide higher gain compared to the Single Input Single Output (SISO) classical systems. In [61], the authors mentioned that resource allocation is a critical challenge for MIMO systems, and it has considerable effect on the expected Quality of Service (QoS). They proposed a method to enhance the resource

allocation for an uplink cooperative MIMO OFDMA multiuser system.

In [62], the authors mention that a proper solution to increase capacity in the future mobile network is maintaining multiple wireless connections. This means that a UE is capable to use multiple radio resources of different serving eNBs simultaneously, and potentially employ carrier aggregation over all those resources. However, the author motioned that most of the spectrum aggregation methods are applicable in the downlink, where an eNB is responsible for radio transmission and power availability is not an issue. They investigated the advantages of allowing for decoupled associations in the dual-connectivity scenarios, where UEs can use radio resources of two serving eNBs for the uplink.

In [63], the authors mentioned that the Pattern Division Multiple Access (PDMA) is a promising multiple access schema for the 5G systems. This method can enhance the spectral efficiency by differentiating signals of users sharing the same resources. The authors claimed that their proposed method, which is an iterative detection and decoding algorithm for PDMA uplink system, offers high performance with reasonable level of complexity. In another study [64], Sparse Code Multiple Access (SCMA) is introduced as a competitive candidate method for 5G systems. The authors mentioned that this technique supports the high throughput as well as the low latency. They investigated different studies about the analysis of the uplink SCMA, and they proposed a low complexity iterative algorithm to maximize the average sum rate of the SCMA systems.

In [65], [66] and [67], the authors discussed about the ever-increasing data demand of the cellular network users as well as the cellular network operators effort to enhance the users' experience from the mobile networks. They also stated that in the term of the dynamic of the cellular data networks, downloads dominate uploads. However, the users attitude has changed during these years, and the huge gap between the uploads and downloads is being filled gradually [68]. Therefore, the users upload will be a more serious challenge for the cellular networks compared to what it is now. To overcome to this challenge, the authors of these papers focused on a popular solution that has been

proposed for the 5G networks. They mentioned that one of the ways to increase the user upload performance is employing the offloading techniques for the cellular networks through WiFi Access Point (AP). They proposed a resource allocation approach for uplink offloading with IP Flow Mobility that is based on weighted proportional fairness for the WiFi access and on linear pricing for the LTE access. One major drawback is that the fairness of this type of algorithms depends on the UEs behavior. Therefore, this type of methods should include a mechanism to confront with the UEs misbehavior or the unreliable UEs. In [65], the authors also proposed a reputation based reaction method to combat the malicious operation.

In [69], the authors mentioned that the capacity crunch caused by the continues growth in cellular data traffic demand is one the major challenges of the current and future mobile networks. They discussed that the deployment of HetNets can be a suitable solution to address the mentioned problem in mobile networks. Moreover, they looked into the D2D communication as another decent approach that can provide further gain for the performance of mobile networks. Finally, to deal with ever-increasing data demands in mobile networks, they proposed a joint uplink resource-scheduling algorithm based on approximate dynamic programming. They applied their approach into a heterogeneous network that supports D2D communication, and they showed that in this situation their approach offers considerable gain to the mobile network.

In [70], the authors mentioned that the conventional mobile devices were mostly used to consume the multimedia content. However, the advances in different aspects of the science and technology caused an evaluation in those devices capabilities and introduced a new generation of the mobile hardware with advanced applications that generate multimedia content. This means that the high data rate services for the uplink will be an inevitable demand of the mobile networks users. Beside this, the cloud-based services make this demand a more serious challenge. The authors of [70] discussed that the methods such as Multi User MIMO (MU-MIMO) that enhance the LTE uplink spectral efficiency are promising approaches to address this problem. LTE-Advanced uses the clustered Single Carrier FDMA (SC-FDMA) approach as the uplink multiple access

schemes. However, there are some constrains like contiguity constraint and robust rate constraint that affect the system performance. These constraints make the usage of the MU- MIMO techniques more complicated. In this, paper the authors proposed a MU_MIMO uplink scheduling with considering the mentioned constraints. They mentioned that the proposed method aims at examining the resource allocation in time, frequency, and spatial dimensions to users.

In [71], the authors refer to the energy saving as one of the highest priority challenges for the future mobile networks. In their idea, the energy consumption of the battery driven devices should be reduced. They claimed that without new approaches to limit the energy consumption of the mobile network devices, there is a high potential that the users of the future wireless networks will be searching for power outlets rather than network access. They mention that to avoid such a problem there is a clear need for the methods that increase the battery lifetime as well as energy efficient approaches (so-called green wireless communication technologies) to reduce the energy consumption of the mobile network devices including eNBs and UEs. They introduced CoMP and D2D as two key technologies to achieve the green wireless communication. However, the usage of these two technologies together in LTE-Advanced mobile networks raises a new challenge. In the LTE-advanced mobile networks, the orthogonality of the subcarriers solves the intra-cell interference problem, but this orthogonality will be lost when D2D communication occur among the cellular users. In [71], the authors used CoMP to propose a scenario, in which the inter cell interference between the eNBs as well as the intra cell interference between the CU and D2D have been removed.

So far, in this section, we discussed the different methods that were proposed to improve the user upload performance as well as the mobile network performance. Some of those proposed methods require users' cooperation for the uplink process. This issue raised other important problems such as why an UE should participate in other UEs upload process and what are the challenges. Beside the technical problems such as security issues that the network operator should provide a solution for them, an important issue needs to be investigated. This issue is about how to motivate users to join the cooperative

uploads to assist in the other UEs upload process. Some studies tried to deal with this problem, and we are going to review them in the following.

In [72], the authors discussed that the rapid growth of the usage of the smart phones and tablets with various data intensive applications leads to huge amount of data traffic over the mobile network. Some of the proposed solutions to this problem such as widening the channel bandwidth are useful up to some extent; however, they impose many costs to the mobile networks. Therefore, the focus should be on the more economical solutions that can deal with this problem. The authors mentioned that offloading part of the cellular traffic over other coexisting networks is a promising solution to face this problem. However, in this type of solutions we need to encourage the users to participate in the offloading schema. To do so, we need to motivate the users to subscribe to offloading service. In [72], the authors used contract theory to model delayed offloading process as a monopoly market. In this model, the operators try to set up optimal quality-price contract and suggest it to users, and each user picks a proper contract item to maximize its own utility by comparing the alternatives. Moreover, they proposed an incentive mechanism to encourage users to exert their delay and price sensitivity in exchange for service cost.

In [73], the authors mentioned that numerous numbers of people use smart phones, and this phenomenon encourages the applications to employ the power of the smart phones users' collaborations. These applications can be divided in two groups: data acquisition and distributed computing. In both groups, there is a master that wants to use the UEs' collaboration. In the distributed computing applications, a master intends to solve a complex problem using distributed computation power of the UEs. In the data acquisition application, a master gathers data from the UEs to make a database. The authors of [73] proposed the incentive mechanisms that offer reward to the users to encourage them to collaborate in the both types of applications. They discussed several scenarios in this article. For the data computation applications, they employed contract theory to investigate how a master rewards different types of users. In case of the data acquisition applications, they used reward-based collaboration for the master in order to motivate more users and releasing less reward.

As seen in some of the related work, the authors proposed to let the owner of the data file (owner UE) to communicate with another neighbor UE and ask it to upload the data file in behalf of the owner UE. In these methods, the owner UE sends its data to another neighbor UE (that has a better communication channel with the serving eNB) over a D2D communication channel. After that, the neighbor UE acts as a relay for the owner UE and it forwards the data to the serving eNB using the mobile network upload resources. In all these methods, the UEs only communicate with their serving eNB.

The main differences between these works and one of our proposed methods (UUC) are the way we use other UEs upload resources as well as using a coordination set of eNBs in the upload process of the cell-edge users. In the UUC method, the owner UE starts the upload of a data file by dividing the data file into a number of pieces and sending those pieces to the eNBs of its coordination set. At a same time, the owner UE asks for the help of neighbor UEs and allows them to upload some of those pieces in behalf of the owner UE. In fact, in the UUC method, we upload different pieces of a data file by using the upload resources of multiple UEs (an owner UE and its helper UEs) at a same time.

2.6 Discrete Event System Specification (DEVS)

DEVS is a formal framework for modeling and simulation. It is based on system theory concepts. DEVS theory provides a precise methodology for representing models, and it presents an abstract description of the system of interest. It supports a formal background for modeling both discrete and continuous systems. According to DEVS formalism, a real system can be defined as a composition of atomic and coupled components. This composition has a hierarchal nature. Atomic models are the basic blocks and a set of one or more interconnected atomic models can form the coupled models. In addition, a coupled model itself can be composed of atomic or coupled models [74][75].

A DEVS atomic model is formally specified by: $M = < X, Y, S, \delta_{int}, \delta_{ext}, \lambda, t_a >$, Where $X = \{(p, v) \mid p \in IPorts, v \in X_p\}$ is the set of inputs events, where $IPorts$ reveals the set of input ports and X_p shows the set of values for the input ports. $Y =$

$\{(p, v) \mid p \in OPorts, v \in Y_p\}$ is the set of outputs events, where OPorts reveals the set of output ports and Y_p shows the set of values for the Output ports. S is the set of sequential states. $\delta_{int}: S \rightarrow S$ is the internal state transition function. $\delta_{ext}: Q \times X \rightarrow S$ is the set of external transition function where $Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$ and e is the elapsed time since last transition function. $\lambda = S \rightarrow Y$ is the output function and $ta: S \rightarrow R_0^+ \cup \infty$ is the time advance function [75].

The above definition means at any given time, a DEVS model is in a state $s \in S$ and it remains in that state for a lifetime defined by $ta(s)$, unless an external event occurs. When the state duration expires, $e = ta(s)$, the model will send the output $\lambda(s)$ through the desired output ports and then it performs an internal transition function to determine the new state by $\delta_{int}(s)$. On the other hand, a state transition can also happen due to the arrival of an external event. In this case, the external transition function determines the new state, given by $\delta_{ext}(s, e, x)$ where s is the current state, e is the elapsed time since last transition and $x \in X$ is the external event that has been received. The time advance function $ta(s)$ can take any real value from the defined interval in the definition. A state with $ta(s) = 0$ is called a transient state which will lead to an instantaneous internal transition. Also if $ta(s) = \infty$, the state is said to be passive such that the system will remain in this state until receiving an external event. It is worth mentioning that the last situation can be used as a termination condition [75].

A DEVS coupled model is formally specified by: $CM = < X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC, select >$, where D is the set of components name. M_d is a DEVS model. EIC is the set of external input couplings. EOC is the set of external output couplings. IC is the set of internal couplings. select is the tie breaker function in case of simultaneous internal event among the components of the coupled model [75].

Figure 2.8 shows an example of DEVS models, which includes one coupled model and three atomic models. This model has two levels: at the first level there is one coupled model (C1, known as the top model), and at the second level there are three atomic

models (A1, A2 and A3). This model has one external input port, and two output ports. The external input port is connected to the input port of A1. One of the two output ports of A1 is connected to the input port of A2, and the other one is connected to the input port of A3. The rest of the interconnections can be seen in Figure 2.8.

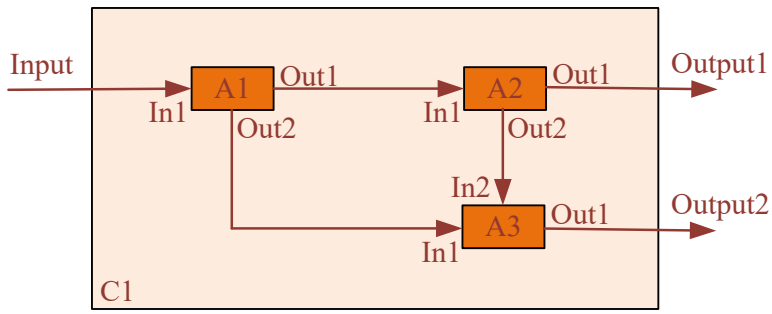


Figure 2.8: A coupled model with three atomic models

The CD++ toolkit provides a framework for programming DEVS models. A model file is used for defining the DEVS coupled model hierarchical structure. A header file is used for defining atomic models as a class, including ports, variables and state definitions. Users can implement definitions of functions such as $\delta_{int}, \delta_{ext}$ and λ in the CPP file, according to the C++ programming language convention. Therefore, the behavioral of the systems is presented through implementation of the atomic models.

3 Shared Segmented Upload

Mobile network operators always investigate new methods to improve the performance of their networks, thus increasing the users' quality of experience. One of the ways that they use to address these challenges is to introduce the new standards, more advanced algorithms and techniques.

In this chapter, we present an advanced algorithm that addresses the problem of large files upload for cell-edge users'. The limited availability of bandwidth in a single communication link between a UE and its serving eNB reduces the data rates, particularly for cell-edge users where the reception/transmission is weak. The Shared Segmented Upload (SSU) algorithm tries to mitigate these issues by spreading the data transfer over a number of the eNBs that participate in a CoMP coordination set.

3.1 The Shared Segmented Upload (SSU) Algorithm

As mentioned earlier, the Shared Segmented Upload (SSU) algorithm tries to improve the upload speed by allowing UEs to spread their data transfer over multiple links to a number of the eNBs in a distributed CoMP architecture, rather than only communicating with the serving eNB. The non-serving eNBs send the pieces they receive from the UE to the serving eNB through the X2 backhaul links. The SSU algorithm has common points with the BitTorrent protocol [10], which is used to speed up the download of large files on the Internet. BitTorrent allows users to join a swarm of hosts to download and upload from each other, simultaneously. BitTorrent is an alternative to the single source, multiple mirror sources technique for distributing data, and can work over networks with lower bandwidth. We adapted this technique to improve data upload from an UE to a set of eNBs. This technique can solve the bottlenecks caused, for instance, by users uploading large files from the UE to the network, improving the upload performance and quality. In brief, we transfer large files in small segments from a single UE to the eNBs in a coordination set, allowing for faster and more efficient transfer of a file, since file

segments are transferred independently, which allows for dynamic adjustment of the data flow in which eNBs with stronger reception receive more segments. Eventually, the collected segments are gathered and organized in the serving eNB, like the pieces of a puzzle.

Let us assume that there is a cell-edge UE (UE1 in Figure 3.1) that wants to upload a large data file. There are three eNBs in its range including eNB1, eNB2 and eNB3, which form a coordination set to support UE1 data file upload.

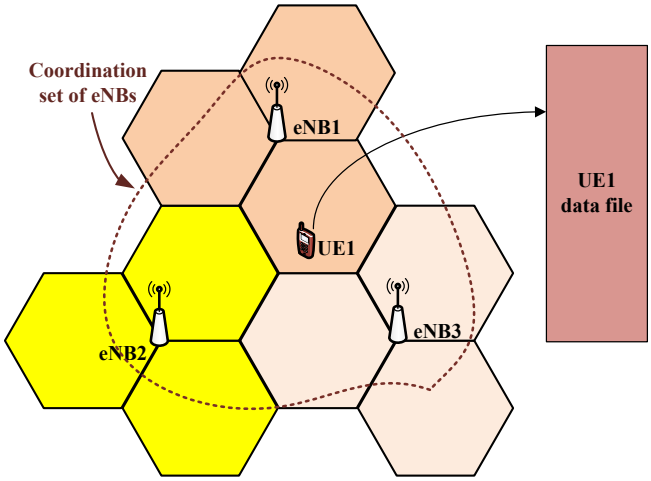


Figure 3.1: A UE with three eNBs in its coordination set

Before the SSU upload process begins, the data file that the UE intends to upload is fragmented into a number of pieces (Figure 3.2) and the UE creates and uploads a file descriptor, *MetaInfo*, based on these pieces. The piece size is usually an exponent of two, and it is selected based on the file size. There is a trade-off between the piece size and the efficiency of SSU. A large piece size makes SSU less effective, as it becomes similar to uploading the large file using traditional techniques; on the other hand, a very small piece size will result in a very large *MetaInfo* message, increasing the overhead. The optimal piece size depends on a number of factors, such as the number of eNBs involved in the CoMP coordination set, and the number of handovers expected to happen during the file transfer. Therefore, the piece size varies depending on the conditions of the uplink channels, which can be adjusted in different simulation scenarios to be investigated for

each situation. In the BitTorrent protocol, the most common piece sizes used are 256 KB, 512 KB, and 1 MB [10]. All file pieces are of equal length, except for the final piece, which is irregular. The number of pieces is determined by dividing the total length of the file by the piece size.

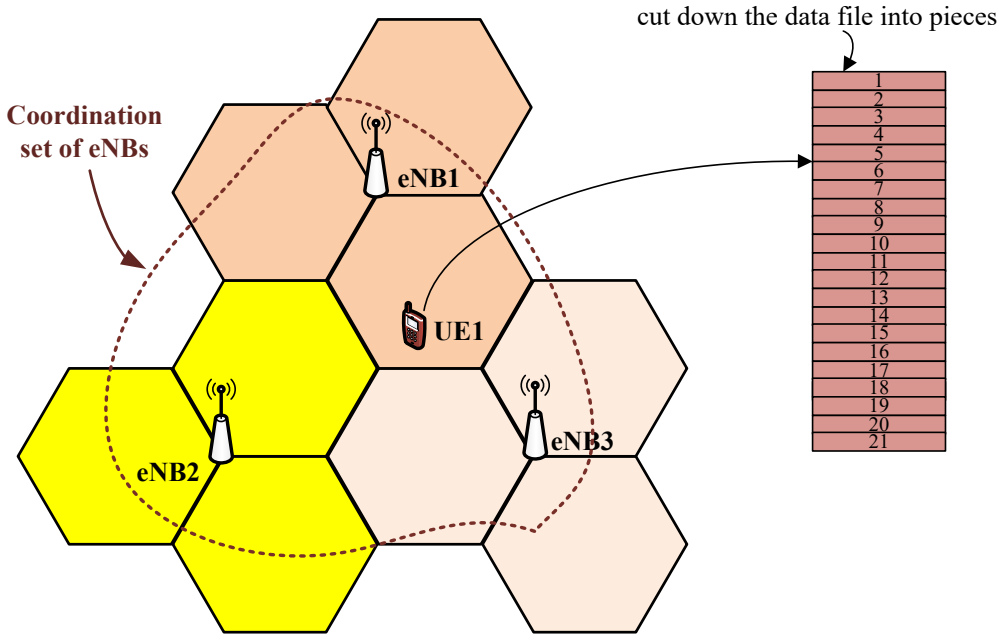


Figure 3.2: The UE creates pieces from the data file

The SSU starts the actual upload process by sending an *Upload Request* message to its coordination set (eNBs that support UE upload jointly). If these eNBs want to allocate resources to this upload process, they send a *Handshake* message to the UE. Upon receiving the *Handshake* message from the eNBs of the coordination set, the UE sends the *MetaInfo* message to the serving eNB, and the serving eNB of the UE is responsible to share this MetaInfo file with the non-serving eNBs of the UE coordination set. After that, the serving eNB transmits the *Bitfield* message to the UE to confirm the reception of the *MetaInfo* message and informs the UE about the available pieces, if there is one. In this manner, the UE does not require to upload those pieces again.

As the next step, the UE starts to upload the pieces to the network. Once the non-serving eNBs receive the pieces they forward them to the serving eNB through the backhaul (X2

links). Finally, when the UE finishes the transmission of all pieces, it sends a *Done* message to the coordination set. Upon receiving the *Done* message, the serving eNB transmits a *Bitfield* message to the UE to inform the UE about the reception status of all the pieces. If there is any missing piece, the UE transmits that piece again, and it sends another *Done* message after that. If the serving eNB confirm the correct reception of the missing pieces (within another *Bitfield* message) the upload process in considered completed, otherwise, the upload is canceled. If the UE insists to upload the data file, it should start this process from beginning. Figure 3.3 represents the SSU steps.

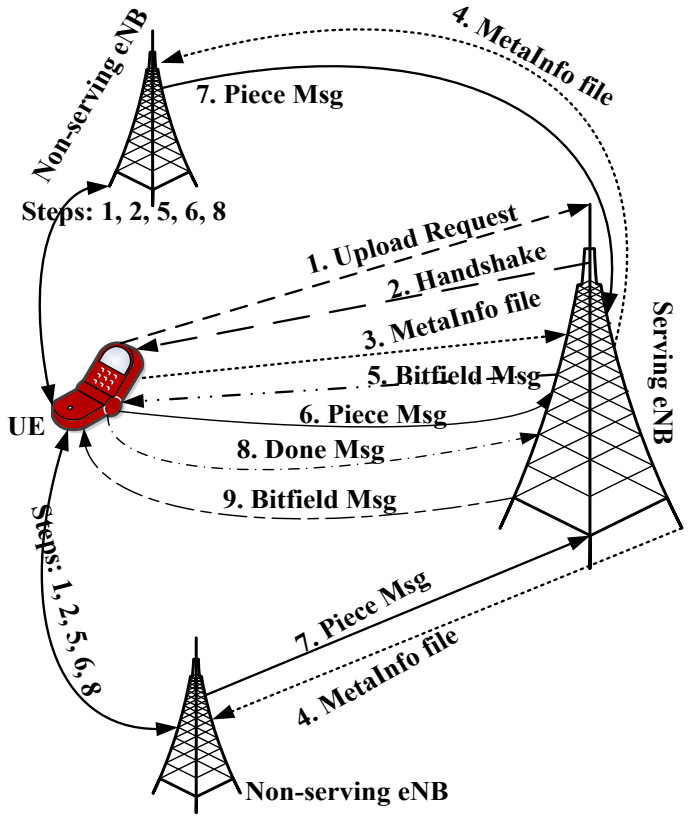


Figure 3.3: Shared Segmented Upload steps

These steps can be summarized as follows:

- **i.** The UE sends an *Upload Request* message to all the eNBs in its CoMP set.
- **ii.** The eNBs reply by sending a *Handshake* message.
- **iii.** The UE sends the *MetaInfo* message to the serving eNB.
- **iv.** The Serving eNB forwards the *MetaInfo* message to other eNBs in the CoMP set.

v. The eNBs acknowledge the receipt of this file by sending the *Bitfield* message, which also tells the UE about the pieces available on the eNBs.

vi. The UE sends the pieces by sending the *Piece* message to all the eNBs in its CoMP set.

vii. The non-serving eNBs send the received pieces to the serving eNB by using the *Piece* message, once they receive them.

viii. The UE stops the data transfer by sending the *Done* message, as soon as all the pieces are sent.

ix. The Serving eNB acknowledges the correct reception of the pieces by sending a *Bitfield* message. If the *Bitfield* message does not acknowledge the reception of all the pieces and the UE has sent the *Done* message once, the UE continues sending the missing pieces until completion, and repeats from step 6. Otherwise, if the *Bitfield* message does not acknowledge the reception of all the pieces and the UE has tried the retransmission process of missing pieces before, it terminates the current upload process (and if it is required, starts new upload process for the same data file from step 1).

3.2 The SSU Algorithm Messages Definition and Structure

In the following sections, in all the messages structure, the first field ('message id' field) shows the type of the message. This field helps the receiver to understand what the different fields of this message are, and how the receiver should deal with them. In addition, the second and the third fields of the messages structure are the destination (receiver) and the source (sender) of the messages respectively. However, the second and the third fields' name may be different in the various messages, but their concept is same.

3.2.1 MetaInfo Message

The UE that wants to upload a data file should create a *MetaInfo* message, and send it to the serving eNB. Each piece is identified by a SHA1 hash code generated from the data contained within that piece. These hash values are each 20 bytes long, and they are concatenated together to form the pieces value dictionary in the *MetaInfo* message. In the upper layers of LTE protocol at the receiver side (eNBs of the coordination set), the hash

value of each piece is used to check the correct reception of that piece. Table 3.1shows the *MetaInfo* message structure. The 'data file name' and 'data file size' fields show the name and the size of the UE data file in order. The 'piece size' shows the length of the pieces (the last piece of the data file may have different size).

	Field Name	Field Type
1	message id	integer
2	serving eNB id	integer
3	sender UE id	integer
4	data file name	string
5	data file size	float (in Byte)
6	pieces size	float (in Byte)
7	pieces hash value	string consisting of concatenation of all 20 byte SHA1 hash values

Table 3.1: The structure of the 'MetaInfo' message

3.2.2 Upload Request Message

The UE starts the actual upload process by sending *Upload Request* message to the eNBs that are involved in the CoMP set (coordination set) of the UE. Table 3.2 shows the *Upload Request* message structure. The 'serving eNB id' represents the id of the serving eNB of the UE. If it is required, the non-serving eNBs use this information to contact with the serving eNB of the UE.

	Field Name	Field Type
1	message id	integer
2	eNB id	integer
3	sender UE id	integer
4	data file name	string
5	data file size	float
6	serving eNB id	integer

Table 3.2: The structure of the 'Upload Request' message

3.2.3 Handshake Message

After receiving an *Upload Request* message, an eNB evaluates its status to see if it can help the UE that has sent the request message. The eNBs of the CoMP set of the UE use this message to inform the UE about their support over the upload of the data file. Table 3.3 represents the *Handshake* message structure.

	Field Name	Field Type
1	message id	integer
2	UE id	integer
3	eNB id	integer
4	data file name	string

Table 3.3: The structure of the 'Handshake' message

3.2.4 Bitfield Message

Upon receiving the *MetaInfo* message, the *Bitfield* message (Table 3.4) is sent by the eNBs of the CoMP set to the UE. *Bitfield* message is also exchanged between the eNBs to coordinate the availability of the pieces (the non-serving eNBs send it to the serving eNB). In this case, instead of copying an UE id in the second field of the *Bitfield* message, it is filled with the address of serving eNB of the CoMP set (eNB-to-eNB *Bitfield* message has not shown in Figure 3.3). The 'piece id' field represents the id of the pieces that have been received at eNBs side so far. The *Bitfield* message size is variable in length, depends on the number of the pieces that have been successfully received.

	Field Name	Field Type
1	message id	integer
2	UE id	integer
3	eNB id	integer
4	data file name	string
5	pieces id	array of integers

Table 3.4: The structure of the 'Bitfield' message

3.2.5 Piece Message

The *Piece* message carries the actual data of the UE's data file. Table 3.5 represents the *Piece* message structure. At the receiver side, the eNBs of the coordination set use the payload part of this message to create the 20-byte SH1 hash key, and compare that with the correspond hash key of this piece in the *MetaInfo* file to double check the correct reception of this piece.

	Field Name	Field Type
1	message id	integer
2	eNBs id	integer
3	sender UE id	integer
4	data file name	string
5	Payload	

Table 3.5: The structure of the 'Piece' message

3.2.6 Done Message

After sending all the pieces of the data file to the eNBs in the CoMP set, the UE send a *Done* message to the CoMP set. This means that it has completed the upload of all the pieces, and there is no more piece to be uploaded. Table 3.6 shows the structure of the *Done* message. In this table, the 'last piece number' filed shows the id of the last piece of the UE data file. The 'eNBs id' field represents the id of the serving eNB of the UE that sends this message.

	Field Name	Field Type
1	message id	integer
2	eNBs id	integer
3	sender UE id	integer
4	data file name	string
5	last piece number	integer

Table 3.6: The structure of the 'Done' message

3.3 Summary

The Shared Segmented Upload (SSU) algorithm introduces an uplink schema for LTE-Advance networks. This method improves the cell edge user's uplink performance by transferring large files in the small segments from a single UE to the eNBs in a coordination set. The core idea of the SSU algorithm has common points with the BitTorrent protocol. Before starting the actual upload, the UE creates a data file descriptor and save the information of the pieces in that file. This data file descriptor acts as a guideline for the eNBs of the coordination set, and let them follow the upload status of the pieces. Finally, the serving eNB gathered all the pieces (the pieces it has received from the UE, and the pieces that non-serving eNBs have forwarded them to the serving eNB), and form the data file.

4 Upload User Collaboration

As discussed earlier, IMT-2000 and IMT-2020 systems need to provide a consistent service for the UEs regardless of their location. As we discussed, providing high data rates to UEs in all coverage areas is challenging, especially when they are located near the cell's border. Different techniques, such as CoMP (introduced in release 11 of 3GPP) improve this by reducing interference and enhancing the signal strength received.

In this section, we discuss an algorithm that improves the cell-edge users' upload in the distributed CoMP scenarios. The Upload User Collaboration (UUC) algorithm uses multiple UEs' upload power to speed up the upload process of a specific UE. Although with some variations, this method can be used for the non-cell edge UEs or in the non-COMP scenarios. In addition, this method can be employed to enhance UEs download process.

4.1 The Upload User Collaboration (UUC) Algorithm

The Upload User Collaboration (UUC) algorithm focuses on enhancing the UEs upload process by using the upload power of multiple users that are close. We assume that these UEs are served by same subset of eNBs (as their coordination set eNBs), but this assumption can be omitted by letting the UEs communicate with different subsets of eNBs. Let us start by presenting the key idea of UUC by using a simplified example. After that, we will explain each step of the UUC in detail. We assume there is a UE in a mobile network (UE1 in Figure 4.1) that wants to upload a data file and there are three eNBs in its range. The eNBs form a coordination set, with eNB1 acting as the serving eNB of this UE. In addition, there are nearby UEs that UE1 can directly communicate with them. UE1 (from now called owner UE) divides the data file into the pieces (Figure 4.2).

The upload process initial steps are similar to those of the Shared Segmented Upload

(SSU) algorithm (with some modifications) [76]. This means that by using the UUC algorithm, UE1 sends an *Upload Request* message (see section 4.2.2) for the eNBs in its range. At the same time, it queries the neighboring UEs in order to see if they are willing to help with the upload process using an *Upload Assistance* message (see section 4.2.3).

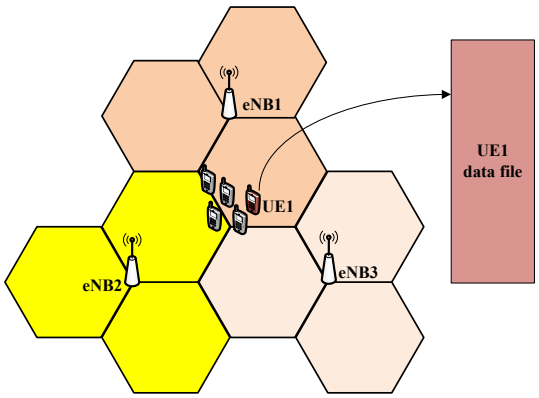


Figure 4.1: A UE with three eNBs in its coordination set

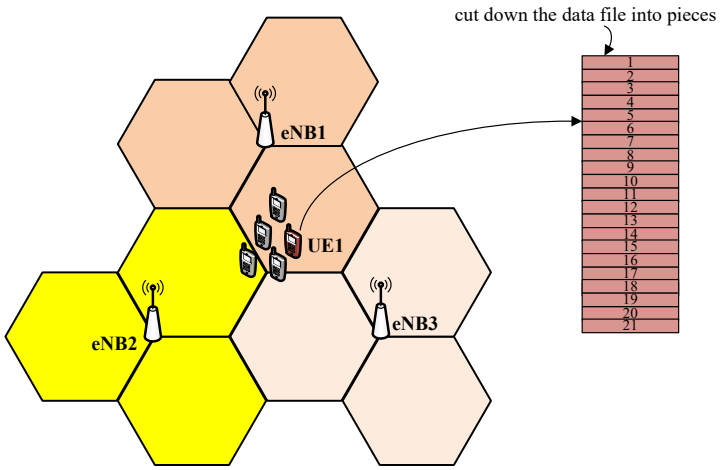


Figure 4.2: The UE creates pieces from the data file

After receiving a *Handshake* (see section 4.2.4) from the eNBs of the coordination set, the owner UE starts to upload the pieces. In addition, if any of the neighbor UEs wants to help (from now on, called helper UEs), they send a *Confirmation* message (see section 4.2.5) to the owner UE. When the owner UE receives this message, it assigns a number of pieces to each of these helper UEs. In our example, two helper UEs want to help the owner UE (Figure 4.3). After receiving their portion of the data file pieces (over the D2D

communication channels), the helper UEs use the SSU algorithm to upload the pieces. To do so, they use their own communication channels with the eNBs in the coordination set, and the owner UE uploads the rest of the pieces (Figure 4.4). This means that we are uploading a data file using three different communication channels (each UE has its own communication channel). In section 2.5, we mentioned some of the works that focus on the challenges that needed to be addressed to encourage a UE (helper UE) to participate in the upload process of another UE (owner UE).

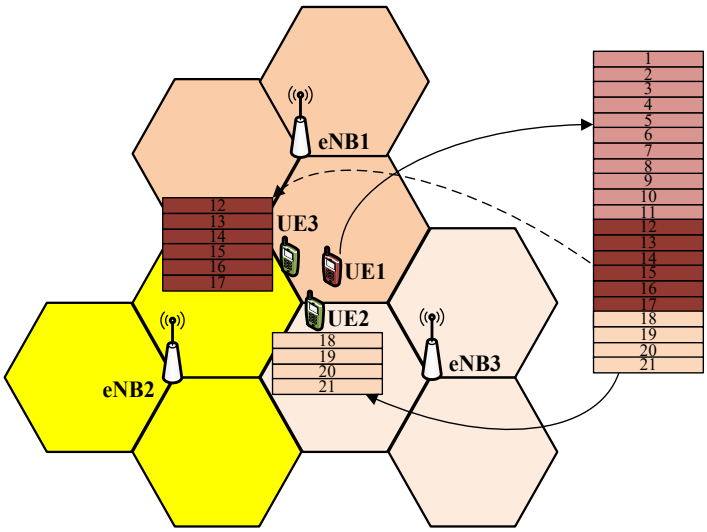


Figure 4.3: The owner UE sends the pieces to the helper UEs

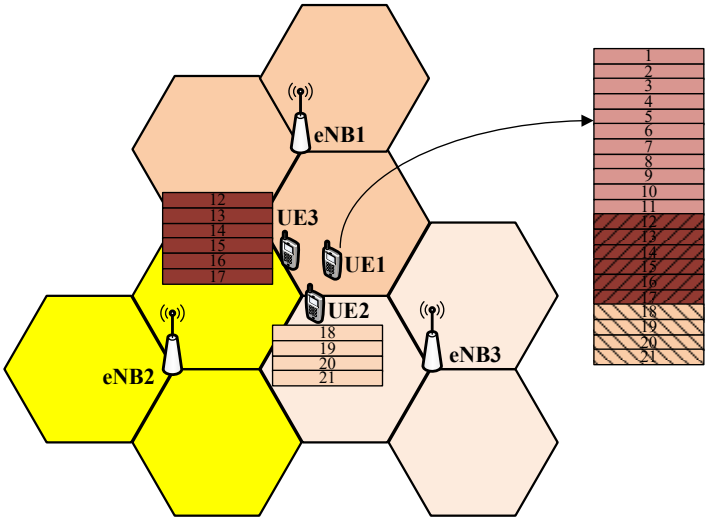


Figure 4.4: The owner UE no longer has to upload the entire file

One may consider a situation in which the file is initially distributed across multiple UE's. Let us assume that the helper UEs have a part (or all) of the data file pieces. Therefore, when they receive an *Upload Assistance* message from the owner UE, they reply with a *Confirmation* message that shows which pieces are already available. The rest of the upload process would be the same as above.

From the eNBs' perspective, they start an upload with the owner UE based on the SSU. During the upload process, they may receive control messages from potential helper UEs showing that they want to assist the owner UE. After performing the required steps to initiate an upload process, the communication channels between these helpers UEs and their supporting eNBs will be established, and they will be able to upload the owner UE pieces as their own pieces. The eNBs forward the received error-free pieces to the Mobility Management Entity (MME), or another entity at a same level as MME. Finally, the MME uses all the pieces to reconstruct the original data file.

Performing a user upload based on these steps means that different pieces of a data file can be uploaded through multiple communication channels. This method tries to speed up the upload process of the UEs regardless of their position in the cell. Therefore, the UUC can be used to enhance the upload process of both cell-edge UEs and non-cell-edge UEs. The steps of the UUC and required messages are described in detail in the following sections.

4.2 The UUC Algorithm Message Definition and Structure

In all the messages structure, the first field ('message id' field) shows the type of the message and helps the receiver to understand what the different fields of this message are, and how it should deal with them. In addition, the second and the third fields of the messages structure are the destination (receiver) and the source (sender) of the messages respectively. Although the second and the third fields' name may be different in the various messages, the concept is same. In addition, the last field of the messages structures shows that if the received messages are an initial transmission or a retransmission of an earlier message. In this case, value '1' means that this message is an

initial transmission and value N, where N is an integer and $1 < N < 5$, shows that this message is a retransmission of a previously transmitted message.

4.2.1 Data File Descriptor Message (DFD)

The owner UE that wants to upload a data file should create a Data File Descriptor (DFD). It sends this DFD to its serving eNB later on after establishing an upload session with the serving eNB (or other eNBs in the coordination set). This message carries information about the owner UE and the data file that it wants to upload, including owner UE id, data file name, and data file size. In addition, some fields have information about the data file pieces: the number of the pieces, and the piece length or size. With respect to this file piece size, three different versions are considered: Uniform Piece Size (UPS), Variable Piece Size (VPS) and Dynamic Piece Size (DPS).

4.2.1.1 Uniform Piece Size (UPS)

In this version, all the pieces of the file are the same size (except for the last piece). In order to determine the piece size, the owner UE at least considers two factors: the total file size, and the quality of the communication channel between the owner UE and its serving eNB. 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, which imposes an increased overhead caused by the additional control messages. On the other hand, a large piece would be difficult to send in a reasonable number of segments, since larger pieces would need to be divided into a large number of segments in the lower layers of the LTE protocol stack. The current assumption for the smallest and the largest piece size regardless of the data file size is 128KB and 2048KB respectively. The piece size can be 128, 256, 512, 1024 or 2048 KB. These assumptions are based on the BitTorrent specification, which states that the piece size is usually a power of two and the most common piece size is 256KB [10]. The owner UE determines the piece size solely. This is because in the initial steps of the upload process when the DFD file is created, the availability of helper UEs is unknown.

In Table 4.1, the ‘sender UE id’ field is the id of the sender of this message, which is the

owner UE. The 'DFD type' field says which method (UPS, VPS or DPS) is used by the owner UE. The 'data file name' and the 'data file size' fields are the name and the size of the data file of the owner UE respectively. The 'piece length' field is the length (size) of the individual pieces. In the UPS method, all the pieces have same size, except for the last piece. Therefore, the first element of the piece length array is the common piece size and the second field is the last piece size. The 'number of the pieces' field is the number of pieces to be uploaded correctly to complete the file. The 'address of the first byte of the pieces' and the 'address of the last byte of the pieces' fields are the addresses of the first and last bytes of the pieces in the data file. If a data file were divided into 100 pieces, the corresponding DFD message would require an array of 100 integers for each of these two fields.

As shown in Table 4.1, the DFD message size is small enough so that the owner UE can easily upload it to its serving eNB. In case of UPS and VPS, this message has 11 fields, which most of them are the integer type. The other fields with other data types require a small amount of memory as well. In case of DPS, the DFD message only includes 7 out of 11 fields of Table 4.1.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	serving eNB id	integer		✓	✓	✓
3	sender UE id	integer		✓	✓	✓
4	DFD type	string		✓	✓	✓
5	data file name	string		✓	✓	✓
6	data file size	float (in MB)		✓	✓	✓
7	piece length	float array		✓	✓	
8	number of the pieces	integer		✓	✓	
9	address of the first byte of the pieces	integer array		✓	✓	
10	address of the last byte of the pieces	integer array		✓	✓	
11	transmission counter	integer		✓	✓	✓

Table 4.1: The structure of the 'Data File Descriptor' message

4.2.1.2 Variable Piece Size (VPS)

In this case, the owner UE considers multiple piece sizes and divides the data file based on these sizes. During the upload process, and after receiving *Confirmation* messages from the helper UEs (see section 4.2.5), the owner UE sends larger pieces to the helper UEs with a larger preferred piece size or a higher upload data rate. The owner UE sends smaller pieces to the helper UEs with lower preferred piece size or lower upload data rate. To do so, the owner UE announces the available piece sizes of the in the *Upload Assistance* message. The helper UEs can then select their preferred piece size based on their current channel conditions by sending a *Confirmation* message. This method uses the same message structure as UPS, where the piece length field is an array of the available piece lengths.

4.2.1.3 Dynamic Piece Size (DPS)

In this method, the owner UE determines the piece size during the transfer process based on a number of criteria. These criteria include (not exclusively) the simultaneous channel quality of the owner UE and the helper UEs. Therefore, the owner UE does not know the number of pieces or their sizes prior to the upload process. During the upload process, the owner UE determines each piece size, for instance, based on the communication channel condition to the eNB at the time of transmitting to the network. In addition, when it wants to send a piece of the data file to the helper UE, it determines the piece size based on the channel condition information provided by the helper UEs.

At the beginning of the upload process (before the UE starts to send the pieces), the owner UE sends a DFD message (DPS version), as defined in Table 4.1, to the eNBs in the coordination set. As seen in the Table 4.1, the DFD message (DPS version) does not have all the defined fields of this table. Note that there is no information about the pieces inside this message structure. During the upload process (when the owner UE starts to create pieces and send them), it populates the DFD message by inserting the information of each piece (piece number, piece size, the first and last bytes address in the data file). At the end of the upload process (after all the pieces are transmitted), the owner UE sends the final populated DFD message (the one that has all the information about all the

pieces) to the eNBs in the coordination set. The serving eNB uses this DFD message to verify the correct transmission of the pieces, allowing it to perform the proper action if an error is detected.

4.2.2 Upload Request Message

After creating the DFD, the owner UE sends an upload request to the eNBs within its range to see which eNBs are willing to participate in the owner UE's file upload. In addition to the legacy LTE feature of a UE requesting resources from its serving eNB, in this thesis the owner UE can also request resources from the cooperating eNBs in the coordination set. Table 4.2 shows the fields in an *Upload Request* message. The same message structure is used in all three methods (UPS, VPS, and DPS).

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	serving eNB id	integer		✓	✓	✓
3	sender UE id	integer		✓	✓	✓
4	data file name	string		✓	✓	✓
5	data file size	double (in MB)		✓	✓	✓
6	transmission counter	integer		✓	✓	✓

Table 4.2: The structure of the 'Upload Request' message

4.2.3 Upload Assistance Message

The owner UE also requests from neighboring UEs if they are willing to assist the owner UE in the upload process. Table 4.3 defines the corresponding *Upload Assistance* message structure for the UPS, VPS and DPS methods. In the case of DPS, the *Upload Assistance* message structure does not require two of the fields of the Table 4.3, since the owner UE does not have information about the pieces prior to the actual upload of the pieces. In this message, the 'receivers' id' field is actually the broadcast address and means that all the neighbor UEs can receive and decode this message if they want. The 'serving eNB id' includes the id of the serving eNB of the owner UE.

4.2.4 Handshake Message

Upon receiving an *Upload Request* message from the owner UE, the serving eNB and other eNBs which will be supporting the UE's upload, send a *Handshake* message (defined in Table 4.4) to the owner UE. A *Handshake* message has same structure for UPS, VPS and DPS. In addition, the non-serving eNBs send a copy of the *Handshake* message to the serving eNB of the owner UE, to notify it about other eNBs that want to support the owner UE's upload process. The 'eNB id' field shows the id of the eNB that sends this *Handshake* message. The 'UE id' field represents the id of the owner UE. After receiving a *Handshake* message from its serving eNB, the owner UE sends the DFD message to the serving eNB.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	receivers' id	integer		✓	✓	✓
3	sender UE id	integer		✓	✓	✓
4	serving eNB id	integer		✓	✓	✓
5	data file name	string		✓	✓	✓
6	data file size	float(in MB)		✓	✓	✓
7	piece length	integer array (in KB)		✓	✓	
8	number of the pieces	integer		✓	✓	
9	transmission counter	integer		✓	✓	✓

Table 4.3: The structure of the 'Upload Assistance' message

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	UE id	integer		✓	✓	✓
3	eNB id	integer		✓	✓	✓
4	data file name	string		✓	✓	✓
5	transmission counter	integer		✓	✓	✓

Table 4.4: The structure of the 'Handshake' message

4.2.5 Confirmation Message

Other UEs in the network can voluntarily, or by instruction from the eNB, listen to the *Upload Assistance* message from the owner UE. If a candidate helper UE is capable of assisting the owner UE, it transmits the *Confirmation* message to the owner UE. As said, by receiving an *Upload Request* message from the owner UE, the serving eNB and optionally, the other non-serving eNBs that are part of the coordination set, send a *Handshake* message to the owner UE. A similar process is used when the neighbor UEs of the owner UE receive an *Upload Assistance* message. Among the neighboring UEs, the ones that choose to assist will send a *Confirmation* message to the owner UE. The decision made by the UEs on whether to become a potential helper can depend on a number of factors, including but not limited to: the communication channel condition, the remaining battery life, security limitations, and service provider's rewards for assisting other UEs. Table 4.5 shows the structure of *Confirmation* message.

	Field Name	Field Type	UPS	VPS	DPS
1	message id	integer	✓	✓	✓
2	receiver UE id	integer	✓	✓	✓
3	sender UE id	integer	✓	✓	✓
4	data file name	string	✓	✓	✓
5	max data size offer to help in upload	double	✓	✓	✓
6	preferred piece size	integer array		✓	✓
7	estimation of upload data rate	integer	✓	✓	✓
8	address of the first byte of the available pieces	integer array	✓	✓	✓
9	address of the last byte of the available pieces	integer array	✓	✓	✓
10	transmission counter	integer	✓	✓	✓

Table 4.5: The structure of the 'Confirmation' message

In all of these methods (UPS, VPS, and DPS), the helper UEs tell the maximum data size that they are willing to upload for the owner UE. The 'max data size offer to help' field in Table 4.5 shows the value of this parameter. In both VPS and DPS, the helper UEs define their preferred piece size ("preferred size field" in Table 4.5) based on the previously

described criteria. In the case of UPS, the piece size is fixed and the owner UE has already determined it. Therefore, in the UPS method, the 'preferred piece size' field is left empty (or it can be filled by a uniform piece size). In Table 4.5, the 'estimation of upload data rate' field shows the contemporary data rate at helper UE side. The owner UE can use this parameter to estimate the amount of time that a helper UE requires to upload a certain amount of data. The 'sender UE id' and the 'receiver UE id' fields are the id of the helper UE and the owner UE, respectively.

Upon receiving the helper UE's *Confirmation* message, the owner UE sends the file pieces, such that each piece size is equal to or less than the preferred piece size defined by the helper UE. Figure 4.5 shows a flow diagram of the initial steps of UUC. We assume that neither the eNBs nor the helper UEs have any part of the data file, but they could have a part of the data file and let the owner UE know which part of the data file is available to them. In this thesis, the values of the two fields of this message (addresses of the first and last bytes of the available pieces) are always empty.

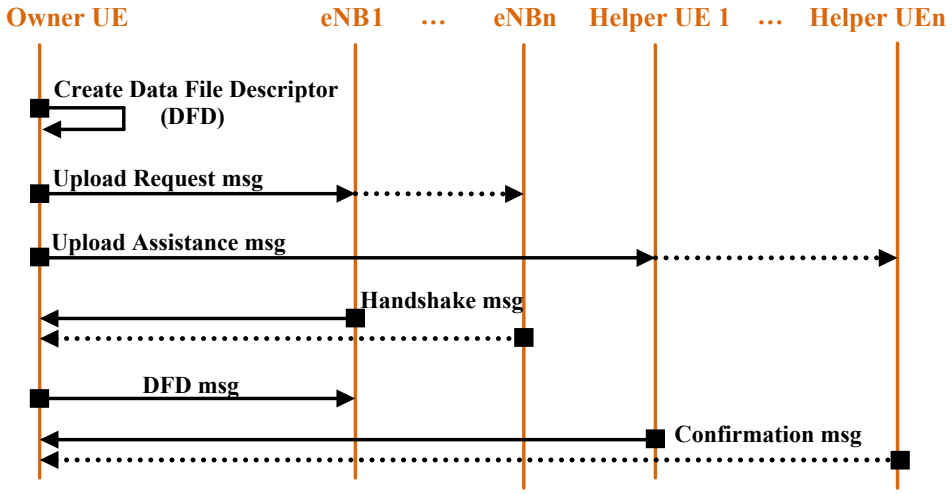


Figure 4.5: Preparation for the pieces upload in UUC (eNB1 is the serving eNB)

4.2.6 Piece Message in an UE-to-UE Communication

By receiving a *Confirmation* message from a helper UE, the owner UE will need to decide if it wants to use the upload power of that helper UE or not. This decision can be

made based on different factors, including the quality of the communication channel between the owner UE and the helper UE, and the number of helper UEs. Upon selecting the helper UEs, the owner UE sends a number of pieces to each of the helper UEs. The number of pieces is determined based on the information in the *Confirmation* message. The owner UE keeps track of the pieces that are assigned to each helper UE (it receives feedback from the serving eNB regarding these pieces). In addition, if the DPS method is being used, the owner UE must update the DFD after creating each piece message. Table 4.6 shows the message structure that the owner UEs use to send pieces to the helper UEs. The 'serving eNB id' tells the id of the owner UE.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	helper UE id	integer		✓	✓	✓
3	owner UE id	integer		✓	✓	✓
4	serving eNB id	integer		✓	✓	✓
5	data file name	string		✓	✓	✓
6	piece number	integer		✓	✓	✓
7	piece size	integer (in KB)		✓	✓	✓
8	address of the first byte of the pieces	integer		✓	✓	✓
9	address of the last byte of the pieces	integer		✓	✓	✓
10	data load			✓	✓	✓
11	transmission counter	integer		✓	✓	✓

Table 4.6: The structure of the 'Piece' message for a UE-to-UE communication

4.2.7 Piece Message in a UE-to-eNB Communication

After establishing the communication channel with the eNBs, both the owner UEs and helpers UEs send their pieces to the eNBs in their range. Table 4.7 defines the structure of the message that the UEs use to send pieces to the eNBs in their coordination set ('receiver eNBs id' field). When the eNBs receive a *Piece* message, they save the required information about the received pieces. For example, the eNBs save the file name that this piece belongs to it, the address of the piece in the data file, the piece number, and

the piece size. This information helps the eNBs avoid future uploading of the same data by other UEs (within a limited period). Note that for the UPS, VPS and DPS methods, there is no guarantee that in future uploads; the UEs will use the same piece length to upload the same data file. Therefore, the piece number is not a good metric to provide information about the available file pieces at the eNBs or helper UEs. The better solution is to use the address of the first and the last byte of the pieces in the data file.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	receiver eNBs id	integer array		✓	✓	✓
3	sender UE id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	serving eNB id	integer		✓	✓	✓
6	data file name	string		✓	✓	✓
7	piece number	integer		✓	✓	✓
8	piece size	integer (in KB)		✓	✓	✓
9	address of the first byte of the pieces	integer		✓	✓	✓
10	address of the last byte of the pieces	integer		✓	✓	✓
11	data load			✓	✓	✓
12	transmission counter	Integer		✓	✓	✓

Table 4.7: The structure of the 'Piece' message for a UE-to-eNB communication

If an owner UE uploads a piece, the 'owner UE id' field and the 'sender UE id' field will be the same in the *Piece* message. Otherwise, if a helper UE uploads a piece then the 'sender UE id' field will be the id of this helper UE. The 'serving eNB id' field is set to the id of serving eNB of the owner UE. The actual data from the data file goes to the 'data load' field. In addition, if the UEs use the DPS method, the owner UE must update the DFD after creating each *Piece* message. It is worth saying that a data piece is a unit of data used in the upper layers of LTE protocol. However, the RLC layer dynamically segments the pieces (that it receives from PDCP layer) into a number of segments (known as RLC PDUs) based on the contemporary communication channel condition.

The MAC layer then uses the RLC PDUs and MAC headers to create transport blocks (TBs), again based on the contemporary communication channel condition. Therefore, the data unit in the lower layers of LTE protocol is a TB. In the rest of this thesis, we will use these definitions.

4.2.8 TB Status Message

Upon receiving a TB of a *Piece* message from the UEs, the non-serving eNBs send a *TB status* message to the serving eNB of the owner UE. This message includes a field that shows the status of the TB reception at the non-serving eNB side. The serving eNB uses this information to avoid unnecessary retransmission request and to determine which one of the eNBs of the coordination set should forward the received piece to the MME (these issue are discussed with more details in the following section). Table 4.8 shows the structure of this message. In this table, the 'TB status' field shows the TB reception situation (0 means NACK and 1 means ACK). The rest of the fields already discussed in previous sections. In another version, based on the request from the serving eNB of the owner UE, the non-serving eNBs send some control information about the received TBs to the serving eNB of the owner UE.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	serving eNBs id	integer		✓	✓	✓
3	non-serving eNB id	integer		✓	✓	✓
4	sender UE id	integer		✓	✓	✓
5	owner UE id	integer		✓	✓	✓
6	data file name	String		✓	✓	✓
7	piece number	integer		✓	✓	✓
8	TB number	integer		✓	✓	✓
9	TB status	integer		✓	✓	✓
10	transmission counter	integer		✓	✓	✓

Table 4.8: The structure of the 'TB Status' message

Figure 4.6 illustrates the sequence diagram when the owner UE sends pieces to the helper UEs. These steps need to be done for each piece. Figure 4.7 and Figure 4.8 show the UUC sequence diagrams when an owner UE or a helper UE uploads a piece respectively.

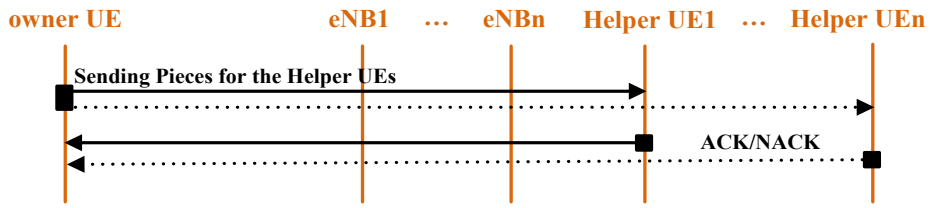


Figure 4.6: The owner UE sends pieces to the helper UEs

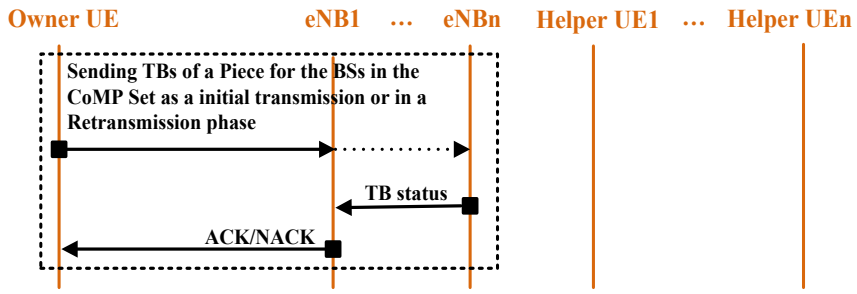


Figure 4.7: The owner UE uploads the pieces (eNB1 is the serving eNB)

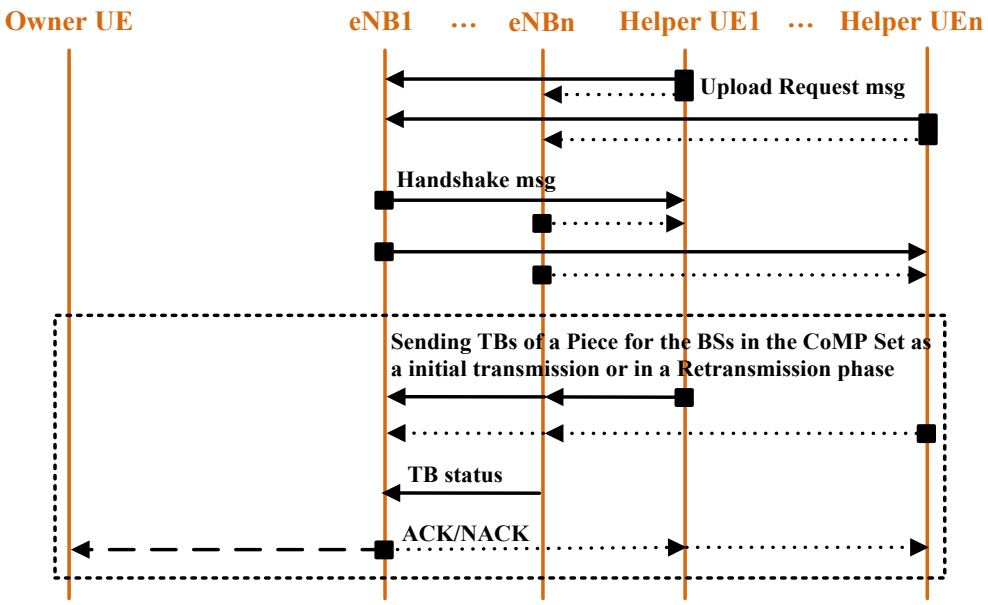


Figure 4.8: The helper UE uploads the pieces (eNB1 is the serving eNB)

4.2.9 Forward to MME Message for the Transporter eNB

The serving eNB is responsible to select an eNB as the transporter eNB. The latter sends a fully received piece to the MME or another entity at the same level as the MME. On the other hand, a piece in the PDCP layer may be divided into a number of TBs in the lower layers of the LTE architecture before it is transmitted. Therefore, one of the following situations may occur when the serving eNB wants to select the transport eNB.

i. The serving eNB has received all the TBs of a piece. Therefore, the serving eNB considers itself as the transporter eNB of this piece, and it forwards the full piece to the MME.

ii. The serving eNB has received some of TBs of a piece, and there is at least one non-serving eNB, which has received all of the TBs of the piece. Therefore, the serving eNB considers that eNB as the transporter eNB for this piece, and it asks that eNB to forward the piece to the MME.

iii. None of the eNBs in the coordination set has received all the TBs of a piece, but the TBs of that piece have been scattered among the eNBs of the coordination set in such a way that collectively they can form the piece correctly. In this case, the serving eNB selects one of the eNBs (this eNB can be the serving eNB itself or one of non-serving eNBs) as the transporter eNB. This eNB is responsible to send the piece for the MME. Other eNBs in the coordination set, including the serving eNB, should provide the missing TBs to the transporter eNB (the serving eNB coordinates this task as we said in previous section). After receiving those TBs, the transporter eNB is able to form the complete piece from the various TBs, and forward it to the MME. There are two proposals for the method used by the serving eNB to select the transporter eNB. The first one is based on the number of the available TBs at the eNB side; the eNB with the most correctly decoded TBs of a certain piece will be the transporter eNB for that piece. The second method can be based on the sum of the sizes of the correctly decoded TBs at each eNB side. In this method, the eNB that has the largest portion of the piece is selected as the transporter eNB. In this case, there is less data transmission over the network backhaul to let the transporter eNB form the whole piece.

iv. None of the eNBs in the coordination set receives all the TBs of a piece and

therefore, the eNBs together cannot form that piece again. As an example, consider a case where all the eNBs missed the same TB. This scenario is discussed in the following sections.

The serving eNB uses the *Forward to MME* message in Table 4.9 to ask the transporter eNB to forward the piece to the MME. If the serving eNB get selected as the transporter eNB of a piece, there is no need for this message. In such a situation, the serving eNB acts as the transporter eNB, and it will forward the received piece to the MME. The 'wait for TBs' field tells if the transporter eNB has the all the TBs of the piece. If this field value is false, it means that the transporter eNB has all the TBs of a piece and it can send the piece to the MME. If 'wait for TBs' value be True, it means that the transporter eNB should expect the missing TBs from other eNBs. The list of those TBs can be found in 'List of TBs' field. Upon receiving that/those missing TB(s), the transporter eNB can send the piece to the MME.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	transporter eNB id	integer		✓	✓	✓
3	serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	String		✓	✓	✓
6	piece number	integer		✓	✓	✓
7	wait for TBs	boolean		✓	✓	✓
8	list of TBs	integer array				
9	transmission counter	integer		✓	✓	✓

Table 4.9: The structure of the 'Forward to MME' message

4.2.10 Missing TBs Message

It is possible that none of the eNBs of the coordination set successfully receives all of the TBs of a piece. However, these eNBs may receive the TBs in such a way that collectively they can form the piece correctly. In the latter case, the serving eNB selects one of the eNBs from the coordination set (the one with the most correctly decoded TBs) as the

transporter eNB. In this case, the transporter eNB (which can be the serving eNB itself or one of the non-serving eNBs) does not have one or some of the TBs of a piece. The serving eNB solves this problem by querying the eNB(s) to determine which eNBs have the missing pieces and requests that they are sent to the transporter eNB. By following this process, the transporter eNB is able to form the whole piece and send it to the MME. We should emphasis that in this situation the serving eNB coordinates all the required communications. This means that the serving eNB figures that out which one of the TBs are required, and then it asks for those missing TBs.

Table 4.10 shows the structure of the message that the serving eNB uses to query the eNBs of the coordination set to send specific TBs to the transporter eNB. The serving eNB may send this message to more than one eNB to query for the different TBs of single piece. In this situation, it inserts the TBs number that it wants from each eNB in the 'missing TBs number' field of their message.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	receiver eNB id	integer		✓	✓	✓
3	serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	transporter eNB id	integer		✓	✓	✓
6	data file name	string		✓	✓	✓
7	piece number	integer		✓	✓	✓
8	missing TBs number	integer array		✓	✓	✓
9	transmission counter	integer		✓	✓	✓

Table 4.10: The structure of the 'Ask for the missing TBs' message

4.2.11 Missing TB Message

If the serving eNB or a non-serving eNB wants to send missing TBs of a piece for the transporter eNB, the *Missing TB* message structure (Table 4.11) is used to forward those TBs to the transporter eNB.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	transporter eNB id	integer		✓	✓	✓
3	sender eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	serving eNB id	integer		✓	✓	✓
6	data file name	string		✓	✓	✓
7	piece number	integer		✓	✓	✓
8	number of the TBs	integer		✓	✓	✓
9	TB list	integer array		✓	✓	✓
10	size of the TBs	double array in KB		✓	✓	✓
11	data load			✓	✓	✓
12	transmission counter	integer		✓	✓	✓

Table 4.11: The structure of the 'Missing TB' message

In this message structure, if the serving eNB of the owner UE were the sender eNB as well, then the 'sender eNB id' and the 'serving eNB id' fields will be the same. The 'number of the TBs' field shows how many TBs were attached into this message. The 'TB list' field shows the sequence number of these TBs and the 'size of the TBs' field tells the size of each the TBs. The 'data load' field includes the concatenation of the TBs. The transporter eNB (which is the receiver of this message) can separate the TBs based on their size and order. It should be reminded that the eNB-to-eNB communication occurs over the X2 links. The latter supports very high data rates communication, which means that the transmission of a number of concatenated TBs can easily be handled.

4.2.12 Piece for MME Message

In all three approaches (UPS, VPS and DPS), the eNBs use the *Piece for MME* message structure in Table 4.12 to forward a piece for the MME. The required information to fill this message is already available in the received piece. Therefore, the transporter eNB just needs to copy and paste them into this message. The 'transporter eNB id' field and the "serving eNB id' field are same if the serving of the owner UE be the transporter eNB

as well. Figure 4.9 pesents a sequence diagram of the UUC when the eNBs send pieces to the MME.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	MME address	integer		✓	✓	✓
3	transporter eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	serving eNB id	integer		✓	✓	✓
6	data file name	string		✓	✓	✓
7	piece number	integer		✓	✓	✓
8	piece size	integer (in KB)		✓	✓	✓
9	address of the first byte of the pieces	integer		✓	✓	✓
10	address of the last byte of the pieces	integer		✓	✓	✓
11	data load			✓	✓	✓
12	transmission counter	integer		✓	✓	✓

Table 4.12: The structure of the 'Piece for MME' message

4.2.13 Renewal Request Message

During the UUC upload process, an owner UE may want to extend its cooperation with a helper UE. In this case, the owner UE sends a *Renewal Request* message to the helper UE to inform it of the remaining pieces that should be uploaded. The owner UE uses the defined message structure in Table 4.13 for the *Renewal Request* message. The owner UE uses the *Renewal Request* message when it estimates that the helper UE is close to finishing the upload of the assigned pieces. The owner UE can make such an estimation based on the total size of the pieces that it assigned to the specific helper UE, and the elapsed time. Moreover, it can use the ACK feedback from the serving eNB (ACK messages of the received pieces show how much data is left at the helper UE that should be uploaded). In Table 4.13, the 'remained data size' field shows the size of data that still needs to be uploaded. The 'pieces length' field represents the size of the remaining pieces. In the case of DPS, this field value is empty (see section 4.1.3).

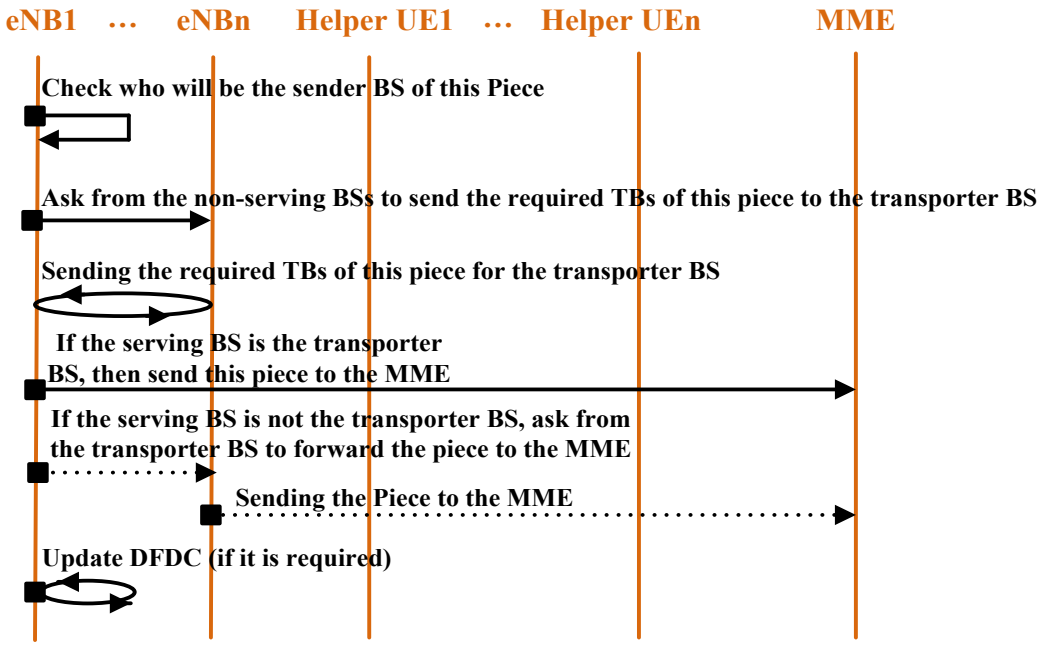


Figure 4.9: Sending a piece to the MME (eNB1 is the serving eNB)

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	helper UE id	integer		✓	✓	✓
3	owner UE id	integer		✓	✓	✓
4	serving eNB id	integer		✓	✓	✓
5	data file name	string		✓	✓	✓
6	remaining data size	integer (MB)		✓	✓	✓
7	pieces length	integer array (KB)		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 4.13: The structure of the 'Renewal Request' message

Another important factor is the upload data rate of the helper UE. Two approaches help the owner UE to receive this information.

- ✓ The helper UE announces its upload data rate to the owner UE when it sends the *Confirmation* message (although it can change as the time passes).
- ✓ The serving eNB sends a control message to the owner UE at fixed intervals that shows the most recently measured upload data rates of the helper UEs.

4.2.14 Renewal Approval Message

In response to receiving the *Renewal Request* message, the helper UE sends back a *Renewal Approval* message that indicates if it wants to continue the cooperation with the owner UE or not. It uses the message structures in Table 4.14 for the *Renewal Approval* message. In Table 4.14, the 'renewal' field has a Boolean type. If the helper UE is willing to extend its cooperation with the owner UE, the helper UE sets it to true. Otherwise, it is set to false. The 'max data size offer to help in upload' field represents the amount of data that the helper UE is willing to upload on behalf of the owner UE. In the case of VPS, the helper UE must select the preferred piece sizes based on the provided information in the received *Renewal Request* message. In the case of DPS, the helper UE chooses a preferred piece size based on its communication channel condition with the serving eNB (or the eNBs in the coordination set). If the helper UE finishes the upload of the pieces that were assigned to it by the owner UE, it may choose to help the owner UE more. To do so, the helper UE can send a *Renewal Approval* message before receiving a *Renewal Request* message from the owner UE. Figure 4.10 represents the sequence diagram of the UUC when the owner UE and a helper UE want to extend the upload cooperation.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	owner UE id	integer		✓	✓	✓
3	helper UE id	integer		✓	✓	✓
4	renewal	boolean		✓	✓	✓
5	data file name	string		✓	✓	✓
6	max data size offer to help in upload	integer (in MB)		✓	✓	✓
7	preferred piece size	integer (in KB)		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 4.14: The structure of the 'Renewal Approval' message

4.2.15 Done Message

After sending all the pieces of the data file to the eNBs in the CoMP set, the owner UE sends a *Done* message (Table 4.15) to the CoMP set to announce that it has sent all the

pieces of the data file. However, sending a *Done* message by the owner UE does not necessarily mean that the upload process has finished completely.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	serving eNB id	integer		✓	✓	✓
3	owner UE id	integer		✓	✓	✓
4	data file name	string		✓	✓	✓
5	last piece number	integer		✓	✓	✓
6	transmission counter	integer		✓	✓	✓

Table 4.15: The structure of the 'Done' message

The upload process can be considered successful when the serving eNB of the owner UE confirms that by sending a message to the owner UE. Therefore, the owner UE should wait for such a confirmation to be assure about the successful data file upload.

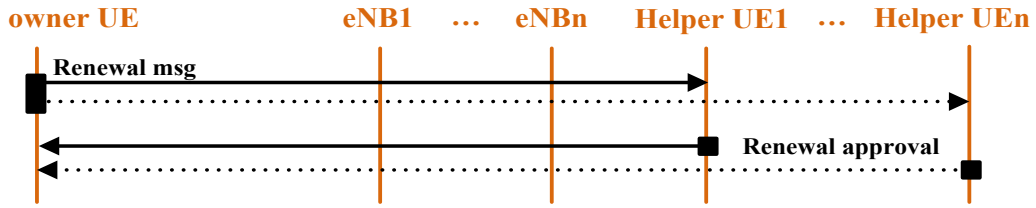


Figure 4.10: Extending the upload cooperation between owner UE and the helper UE

4.2.16 DFDC Message

As described in the previous sections, during the upload process, the non-serving eNBs of the coordination set send the control information of the received pieces to the serving eNB, and the latter inserts the information of each piece into a DFD Complements (DFDC) list. Therefore, the serving eNB has the control information of all the received pieces. The DFDC message structure is shown in Table 4.16. In this table, the piece's status is '1' if it has been received correctly, otherwise it is '0'. The length of the piece number array is equal to the number of the pieces. There is a corresponding piece status

array, which includes the information about the reception of the pieces.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	owner UE id	integer		✓	✓	✓
3	serving eNB id	integer		✓	✓	✓
4	data file name	string		✓	✓	✓
5	pieces number	integer array		✓	✓	✓
6	pieces status	integer array		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 4.16: The structure of the 'DFDC' message

4.2.17 Upload Finished Message

Once the eNBs of the coordination set receive all the pieces of the data file of the owner UE, the serving eNB of the owner UE sends an *Upload Finished* message to the owner UE to let it knows that the upload process is now complete. By receiving this message, the owner UE can be assure that the data file upload has finished successfully. Table 4.17 represents the structure of the *Upload Finished* message.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	UE id	integer		✓	✓	✓
3	serving eNB id	integer		✓	✓	✓
4	data file name	string		✓	✓	✓
5	transmission counter	integer		✓	✓	✓

Table 4.17: The structure of the 'Upload Finished' message

4.2.18 Upload Canceled Message

If the serving eNB wants to terminate the upload process of an owner UE (for any given reason), it uses the message structure in Table 4.18 to inform the UEs about upload

cancelation. The ‘reason code’ filed determines the reason of the upload cancellation. The value of this field refers to the reason that the serving eNB uses to terminate the upload session. The both side of the communication know about the reasons list and their index number.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	UE id	integer		✓	✓	✓
3	serving eNB id	integer		✓	✓	✓
4	data file name	string		✓	✓	✓
5	reason code	integer		✓	✓	✓
6	transmission counter	integer		✓	✓	✓

Table 4.18: The structure of the ‘Upload Canceled’ message

4.2.19 Forward the Received TB Message

If for any given reason the serving eNB wants to ask from the non-serving eNBs to send their copy of a specific received TB, it uses the message structure in Table 4.19. This message includes information about the TB that the serving eNB looking for it. The ‘data file name’, ‘piece number’, and ‘TB number’ fields are used to select the TB.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	non-serving eNB id	integer		✓	✓	✓
3	serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	piece number	string		✓	✓	✓
7	TB number	integer		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 4.19: The structure of the ‘Forward the Received TB’ message

4.2.20 Received TB Message

In response to the *Forward the Received TB* message, the non-serving eNBs send their copy of the said TB by using the *Received TB* message. Table 4.20 illustrates the different fields of this message. All these fields were discussed in previous sections.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	serving eNB id	integer		✓	✓	✓
3	non-serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
6	data file name	integer		✓	✓	✓
7	piece number	string		✓	✓	✓
8	TB number	integer		✓	✓	✓
9	data load					
10	transmission counter	integer		✓	✓	✓

Table 4.20: The structure of the 'Received TB' message

4.2.21 Wait for the RLC Layer Recovery Message

If for any given reason, the MAC layer of the serving eNB cannot recover data from an erroneous received TB even after three retransmissions, it can leave the data recovery responsibility for the upper layer (RLC layer). However, in such a situation, the serving eNB should inform the UE about this decision. The serving eNB uses the *Wait for the RLC Layer Recovery* message (Table 4.21) for this purpose. In this table, the 'piece number' and the 'TB number' fields help the UE to get information about that part of data that none of the eNBs of the coordination set have received them without error.

4.2.22 Wait for the PDCP Layer Recovery Message

In this section, let us assume that the RLC segment size is equal to the piece size. Same as MAC layer and the TBs, it is possible that the RLC layer of the serving eNB cannot recover data from an erroneous received piece even after three retransmissions. Therefore, the serving eNB uses message structure in Table 4.22 (*Wait for the PDCP*

Layer Recovery message) to notify the UE that it wants to leave the data recovery responsibility of a specific piece for the upper layer (PDCP layer). This message has the information about the name of the data file as well as the piece number.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	UE id	integer		✓	✓	✓
3	eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	string		✓	✓	✓
6	piece number	integer		✓	✓	✓
7	TB number	integer		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 4.21: The structure of the 'Wait for the RLC Layer Recovery' message

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	UE id	integer		✓	✓	✓
3	eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	string		✓	✓	✓
6	piece number	integer		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 4.22: The structure of the 'Wait for the PDCP Layer Recovery' message

This message or the one that we discussed in the section 4.2.21 provides more information to the UE about the exact status of the data file upload. In one version, if the communication protocol stack uses an error recovery method in the upper layers of the protocol stack, the UE may decide to continue the upload process and wait for the upper layers decision about the erroneous received piece or transport block.

4.2.23 Error Checking and Retransmission

Error checking is performed in three levels in the UUC algorithm. The receivers (the eNBs in the coordination set of a certain UE) perform the first two error checking steps in the MAC and RLC layers of LTE protocol stack. The serving eNB, and only the serving eNB, performs the last and the highest level of error checking at the end of the upload process in the upper layers of the LTE protocol stack. As said, the MAC and RLC layers are responsible for the first and second levels of error checking respectively. To discuss the retransmission process in the MAC layer, let us recall that the RLC layer dynamically divides the pieces that it receives from PDCP layer into a number of segments known as RLC PDUs. The MAC layer then uses the RLC PDUs and MAC headers to form transport blocks (TBs).

As defined in the previous sections, we assume that there is a number of eNBs that form the CoMP coordination set to support a UE upload. The serving eNB is the only eNB in the coordination set that sends ACK or NACK control messages to the UE. In addition, the non-serving eNBs send a feedback to the serving eNB when they receive a TB. As an example, let us assume that we have a coordination set of three eNBs supporting a UE upload, where eNB1 is the serving eNB, and eNB2 and eNB3 are the non-serving eNBs of the coordination set. When the UE transmits a transport block in the upload process, one of the following two situations (correct reception or retransmission) can happen at the eNB receivers.

4.2.24 Correct Reception

In the following three situations, the serving eNB does not need to send a retransmission request to the UE, because at least one of the eNBs in the coordination set is able to recover data from the received Transport Block (TB).

- ✓ All the eNBs of the coordination set receive the transmitted TB without error. As a result, the serving eNB sends an ACK message for the UE.
- ✓ A subset of the eNBs of the coordination set, including the serving eNB, receives the transmitted transport block without error and the rest of the eNBs receive it with error. In this case, the serving eNB sends an ACK message for the UE.

- ✓ A subset of the eNBs of the coordination set excluding the serving, receives the transmitted TB without error, while the remaining eNBs, including the serving eNB, receive it with error. In this situation, the serving eNB waits for the feedback (*TB Status* message) from the other non-serving eNBs. If at least one of the non-serving eNBs sends a positive acknowledgement for the serving eNB, the serving eNB is able to send an ACK for the UE.

4.2.25 Retransmission

If none of the eNBs in the coordination set correctly recovers data from the received TB, the serving eNB requests a retransmission of that TB.

4.2.25.1 Retransmission Mechanism for the MAC layer

The UUC can use the LTE-Advanced retransmission method. However, we updated the LTE retransmission method to prevent the cell-edge UE from unnecessary retransmission. The detail of this proposed method is discussed in chapter 7.

4.2.25.2 Retransmission Mechanism for the RLC layer

Like in section 4.2.22, let us assume that the RLC segment size is equal to the piece size in this section. The RLC layer tries to recover data from the received piece (At the receiver side, the MAC layer put the received TBs of a piece together to form that piece and deliver the piece to the RLC layer). If the RLC layer recovers the data, it sends an ACK for the received piece. Otherwise, the RLC layer asks for the retransmission of the piece by sending a NACK message. The retransmission process can occur up to three times in this level. After that if the RLC layer cannot resolve the piece correctly, it leaves the error recovery for the upper layers in LTE protocol stack and it sends *Wait for the PDCP layer recovery* message to the UEs. In another version, the helper UEs may receive the NACK message, and they can retransmit that piece as well.

4.2.25.3 Error Checking in the Upper Layers of the Protocol Stack

The third and last level of error checking occurs in the layers above RLC layer. In this

step, the serving eNB checks the received pieces information with the DFD message to make sure that the eNBs of the coordination set collectively received the pieces of the UE data file completely and correctly. After receiving the *Done* message from the owner UE, the serving eNB checks the DFD and the DFDC together in order to see if their information about the pieces is equal. In this case, the serving eNB sends *Upload Finished* message to the owner UE. In addition, the serving eNB sends the DFD to the MME. Finally; the MME is able to reconstruct the data file by using this DFD and the received pieces (Figure 4.11).

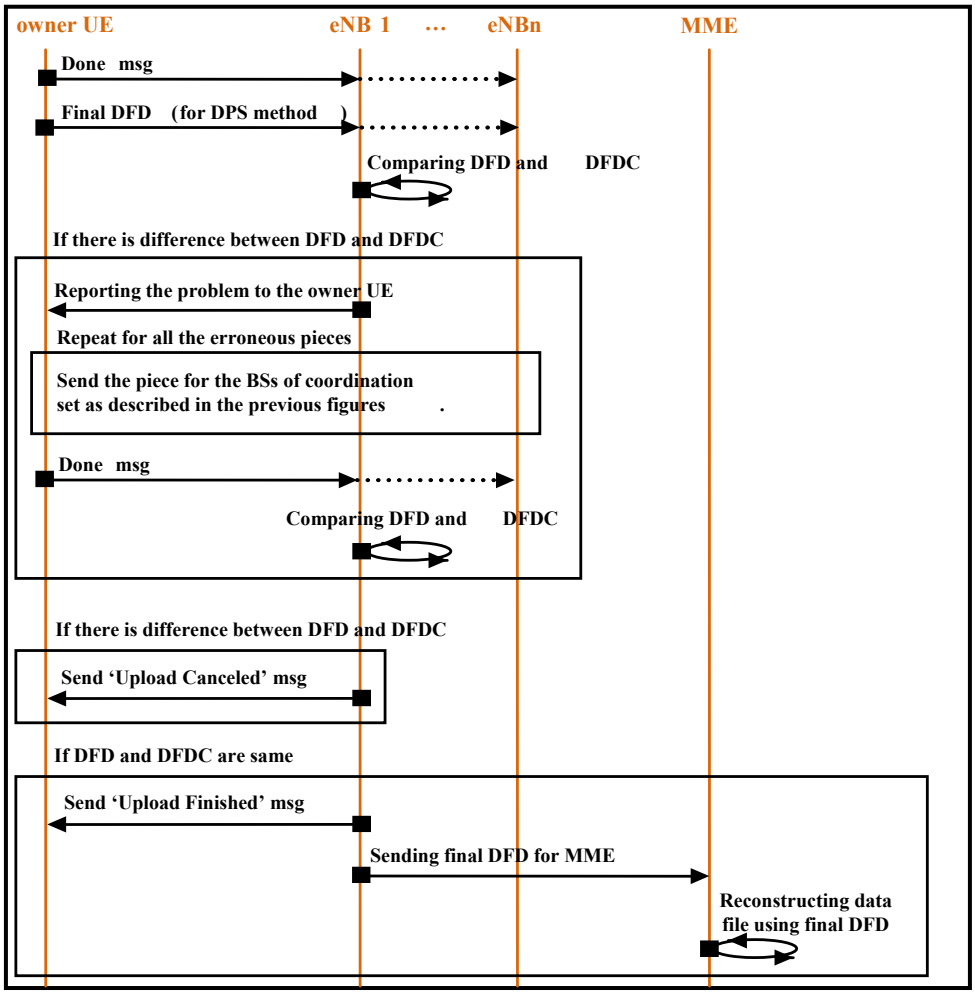


Figure 4.11: Last steps of the UUC (eNB1 is the serving eNB)

If DFD and DFDC are not same, the serving eNB sends a report (DFDC) for the owner UE to let it know about the pieces that have not been correctly received. The owner UE

attempts to resend erroneous pieces based on what we have discussed in the previous sections. After that, the UE sends another *Done* message to the eNBs of the coordination set. If these steps help the serving eNB to resolve the problem then the serving eNB sends the final DFD for the MME. Otherwise, the serving eNB cancels the whole upload process and informs the owner UE that the upload process was unsuccessful. The serving eNB uses the *Upload Canceled* message for this purpose. In the UPS and VPS methods, once the owner UE creates the DFD at the beginning of the upload process, it does not change the DFD during the upload process. Therefore, the serving eNB uses the same DFD file that it received in the primary steps of the upload process to check if all the pieces are received correctly. In the case of DPS, this process is a bit different. When an owner UE uses the DPS method for the upload process, it should update the DFD during the upload process, and it should send the final DFD for the serving eNB after sending the *Done* message.

4.3 Summary

The method proposed in this chapter tries to use multiple UEs to enhance the upload process of a single UE. To do so, the owner UE creates a data descriptor file and shares pieces of the file among the UEs nearby. The owner UE and the helper UEs upload the pieces of file simultaneously. Figure 4.12 shows UUC message transfer steps. Moreover, some important points about the UUC are summarized below:

- ✓ If none of the neighboring UEs wants to help the owner UE, then the owner UE solely uploads the data file.
- ✓ If the helper UE wishes to abort the upload before it is complete, the owner UE will use the feedback received for all the pieces from the eNBs (even the ones that helper UEs upload), and it can upload the remaining pieces itself.
- ✓ The UE continues the rest of its upload, while the eNBs communicate with each other in attempt to recover a received TB.
- ✓ The serving eNB is the only eNB that is responsible to send control messages to the UE.
- ✓ The non-serving eNBs send their feedback about the TBs over the backhaul for the serving eNB automatically. They send TBs if there is a request from the

serving eNB. We can consider different approaches for this step. For example, we can let the non-serving eNBs send their feedback, or even the TBs for the serving eNB, automatically when the error rate in the network is high, since there is a high chance that no other eNBs will be able to recover the data from received TBs. This approach may result in a faster error recovery, but it puts more overhead on the backhaul.

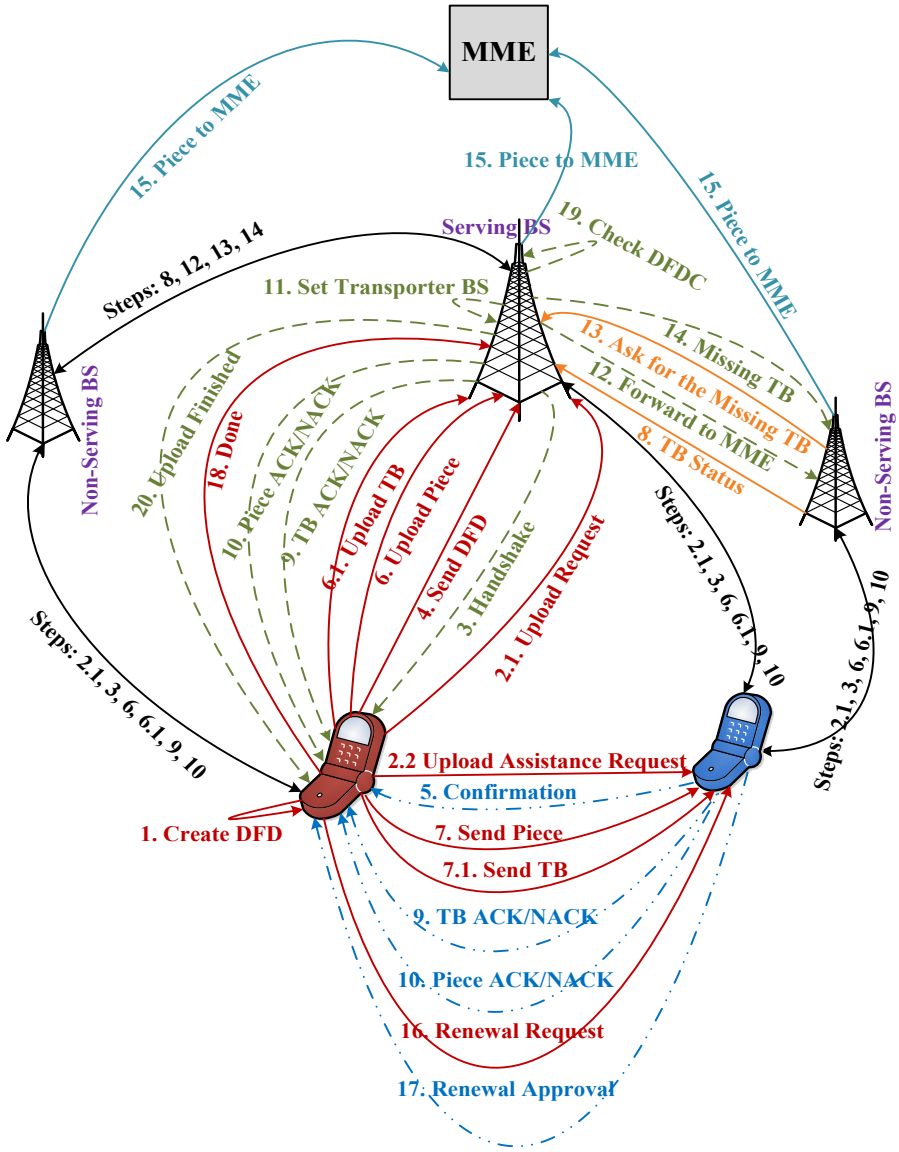


Figure 4.12: Upload User Collaboration algorithm message transfer

5 The Super Set Method

UUC, defined in previous chapters, focuses on enhancing the cell-edge UE upload process by using the upload power of the multiple close users allowing D2D communication between UEs. In UUC, it is assumed that an owner UE and its helper UEs are served by same subset of the eNBs as their coordination set. However, this assumption is not necessarily optimal, and by letting the UEs within the D2D cluster communicate with different subsets of eNBs as their CoMP coordination set can result in additional throughput gains. As an example, consider the situation in which the owner UE shares some part of the data file with the helper UEs and after that one of the UEs (the owner UE or the helper UE) wants to move in such way that this movement causes changes into the UE coordination set. In the previous chapter, we said that, theoretically, a group of the UEs that try to upload a same data file could be either served by the same subset of the eNBs or served by different subsets of the eNBs. However, the proposed method did not provide a solution for the cases where the UEs are served by partially same or completely different subset of the eNBs as their coordination set.

In this chapter, we remove the fixed coordination set limitation and introduce a method to extend the UUC algorithm. This extension improves the UUC in such a way that the UEs that are involved in the upload process of an owner UE can use different coordination set of the eNBs. In other words, we want to let the owner UE and its helper UEs to use different eNBs set as their coordination set. To do so, defining a new concept called *Super Set*, whose basic idea is to handle the upload process of different parts of a same data file by multiple UEs that are using different coordination sets.

5.1 Super Set Types

Let us assume there is an owner UE in a mobile network that wants to upload a data file. In addition, there is at least one helper UE that wants to help with the upload process. If the owner and helper UEs use different eNBs in their coordination set, the serving eNB of

the owner UE can establish a super set to support the UEs upload process. In this method, we can make sure that the UEs remain coordinated during the upload process and they do not upload a piece more than once. To do so, we consider two versions:

- Type I: it is used when the UEs have same serving eNB but they use different non-serving eNBs in their coordination sets.
- Type II: it is used when the UEs use different eNBs as their serving eNB and they may have or have not different non-serving eNBs in their coordination sets.

In both versions, the serving eNB of the owner UE establishes a super set to be able to communicate with other eNBs to gather information about the upload process of the owner UE data file. This information is forwarded to the owner UE to keep the owner UE updated about the upload status of the pieces of the data file.

5.2 Super Set Concept

Before we proceed with the *Super Set* concept, we must discuss two terminologies. When we say that two UEs use same subset of the eNBs as their coordination set, it means that these two UEs have a same group of the eNBs in their coordination set. However, this does not necessarily mean that these UEs have a same serving eNB or the non-serving eNBs in their coordination set as well. As an example, consider the case in Figure 5.1, where both UE1 and UE2 have eNB1, eNB2 and eNB3 in their coordination set. The serving eNB of UE1 is eNB1 and the other two eNBs (eNB2 and eNB3) are the non-serving eNBs of the UE1 coordination set. However, on the other hand, eNB2 is the serving eNB of UE2 and the other two eNBs (eNB1 and eNB3) are the non-serving eNBs of the UE2 coordination set.

When we say that the non-serving eNBs of the coordination set of two UEs are not the same, it means that there is at least one or more non-serving eNB in the coordination set of one of the UEs, which is not a member of the coordination set of the other UE. In our previous example, the non-serving eNBs of the UE1 and UE2 were not the same, i.e. eNB2 is a non-serving eNB member of UE1 coordination set and it is not a non-serving eNB member of UE2 coordination set.

	Serving BS	Non-serving BS
UE1	eNB1	eNB2, eNB3
UE2	eNB2	eNB1, eNB3

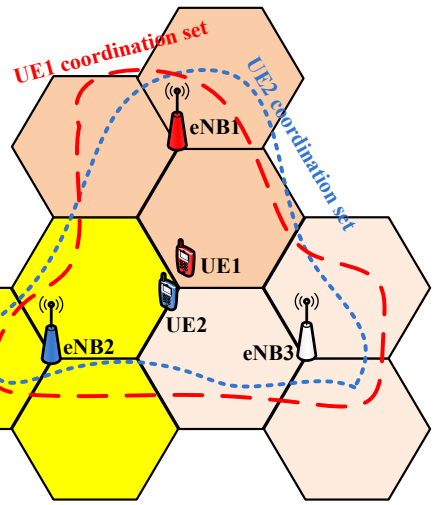


Figure 5.1: Same set of eNBs as the coordination set, but different serving eNBs

5.3 Super Set Definition

From the LTE standard, a subset of eNBs that work together to mitigate inter-cell interference and to increase the UE performance is called a coordination set (CS). Among all the eNBs in the coordination set of a UE, there is one, which acts as the serving eNB and the rest of the eNBs are non-serving eNBs. In this thesis, a group of eNBs is called super set (SS) of a specific UE if they satisfy the following conditions:

- ✓ A super set is comprised of at least two different coordination sets (it means that these coordination sets have different serving eNBs or different set of non-serving eNBs or both) and it inherits the coordination set characteristic.
- ✓ The super set supports the upload process of the group of the UEs in a D2D cluster (i.e. an owner UE and its helper UEs) that implement the transfer of the different parts of a same data file into multiple coordination sets of the eNBs.
- ✓ A super set has at least one serving eNB (sometimes multiple coordination sets may use a same eNB as their serving eNB).
- ✓ Among all the serving eNBs of the super set, one eNB should be able to gather all the information regarding the uploaded pieces to the different coordination sets of the super set. This eNB is called the central serving eNB of the super set and typically is the serving eNB of the owner UE.
- ✓ A super set has only one central serving eNB.

- ✓ Other serving eNBs of the super set are called non-central serving eNBs of the super set.
- ✓ The non-central serving eNBs are the serving eNBs of the coordination set of at least one UE. They are responsible for controlling the upload/download process in their coordination sets and providing feedback for the central serving eNB as well.
- ✓ The non-central serving eNBs feedback is a type of control message.
- ✓ The non-serving eNBs of the coordination sets of a super set are simply called eNB members of the super set
- ✓ The eNBs member of a super set do not need to be neighbors.
- ✓ The number of the eNBs of a super set may be equal or greater than the number of the eNBs of the coordination sets that construct this super set together.

Considering the serving eNBs of the coordination sets, we can have two types of the super sets.

i. Type 1: In a Type 1 Superset, all of the coordination sets of the super set have same serving eNB. - i.e. there is only one serving eNB in the super set. Furthermore, this means that the serving eNB is the central serving eNB of the super set and that there are no non-central serving eNBs in the super set. Since the central serving eNB of the super set is the serving eNB of all the coordination sets that together formed the super set, it has all the information about the uploaded pieces in the different coordination sets of the super set.

ii. Type 2: In a Type 2 Superset, at least there are two coordination sets, which have different serving eNBs. Thus there is more than one serving eNB in the super set. The serving eNB of the owner UE will be the central serving eNB of the super set and the serving eNBs of the helper UEs will be the non-central serving eNBs of the super set. Initially, the central serving eNB of the super set has just the information of the uploaded pieces in its coordination set (for which it is the serving eNB of the owner UE). However, the central serving eNB of the super set needs to gather all the required information of the uploaded pieces in the different coordination sets of the super set. To do so, the non-central serving eNBs (that are

the serving eNB of the other coordination sets of the super set) should send the information of the uploaded pieces in their coordination set to the central serving eNB.

5.4 Owner UE Coordination Set vs. Helper UE Coordination Set

In the context of the super set definition, the usage of coordination sets and super sets in the different scenarios that can occur when a helper UE cooperates with an owner UE, are discussed below. Table 5.1 compares the coordination set of an owner UE and a helper UE and it covers the different scenarios that can occur.

	all eNBs are same	some eNBs are same	serving eNBs are same	non-Serving eNBs are same	our need	SS type
1	F	F	F	F	SS	II
2	F	F	F	T	NR	NA
3	F	F	T	F	NR	NA
4	F	F	T	T	NR	NA
5	F	T	F	F	SS	II
6	F	T	F	T	SS	II
7	F	T	T	F	SS	I
8	F	T	T	T	NR	NA
9	T	F	F	F	NR	NA
10	T	F	F	T	NR	NA
11	T	F	T	F	NR	NA
12	T	F	T	T	NR	NA
13	T	T	F	F	SS	II
14	T	T	F	T	NR	NA
15	T	T	T	F	NR	NA
16	T	T	T	T	CS	NA

Table 5.1: The owner UE coordination set vs. the helper UE coordination set (SS: Super Set, CS: Coordination Set, NR: Not Realistic, NA: Not Applicable)

In Table 5.1, if all the eNBs in the coordination sets of an owner UE and its helper UE

are the same, then the value of the first column is true (T), otherwise, it is false (F). If the coordination sets are partially similar (i.e. they share one or more eNBs, but not all eNBs), then the value of the second column is true, otherwise it is false. If the owner UE and the helper UE have same serving eNB, then the value of the third column is true, otherwise, it is false. If the owner UE and the helper UE have the same non-serving eNB(s), then the value of the fourth column is true, otherwise, it is false. Therefore, these four parameters can form 16 different cases. Table 1 shows all the possible cases, but some of these cases are not realistic. For example, if the first column value is true (all the eNBs of the coordination sets of the owner UE and the helper UE are same), the second column value cannot be false. We discard such cases and we describe the valid cases in more detail. The valid cases of Table 1 can be divided into three main categories:

i. Both UEs (the owner UE and its helper UE) have the same subset of eNBs as their coordination set.

ii. Their coordination sets are partially similar.

iii. They have different coordination sets.

5.4.1 UEs with the Same Subset of the eNBs as Their Coordination Set

This section considers two scenarios in which both UEs have same set of eNBs in their coordination sets.

5.4.1.1 Same Serving eNB and non-Serving eNBs:

This case corresponds to row 16 of Table 5.1. In this case, the same coordination set of eNBs is supporting the upload of both UEs. Considering the fact that the serving eNB and the non-serving eNBs of the owner UE and the helper UE are same, there is no need for extra coordination among the eNBs and the coordination set can support the upload process.

5.4.1.2 Different Serving eNB and non-Serving eNBs:

This case corresponds to row 13 of Table 5.1 in which the owner UE and the helper UE use the same subset of eNBs as their coordination set. However, they have different

serving eNB and furthermore their non-serving eNBs are not the same (Figure 5.1). In this case, a super set Type 2 is required to support the owner UE upload process. This scenario can be illustrated by the example in Figure 5.1. Assume that UE1 is an owner UE and UE2 is a helper UE. Both UEs are supported by same set of eNBs including eNB1, eNB2 and eNB3. In this example, eNB1 and eNB2 are the serving eNBs of UE1 and UE2 coordination sets respectively. Therefore, pairs of (eNB2, eNB3) and (eNB1, eNB3) are the non-serving eNBs of UE1 and UE2 coordination sets in order. According to this example, UE1 super set will have two serving eNBs (eNB1 and eNB2) and one non-serving eNB (eNB3). eNB1, as the serving eNB of the owner UE will be the central serving eNB of the super set and eNB2 will be the non-central serving eNB of the super set (Figure 5.2).

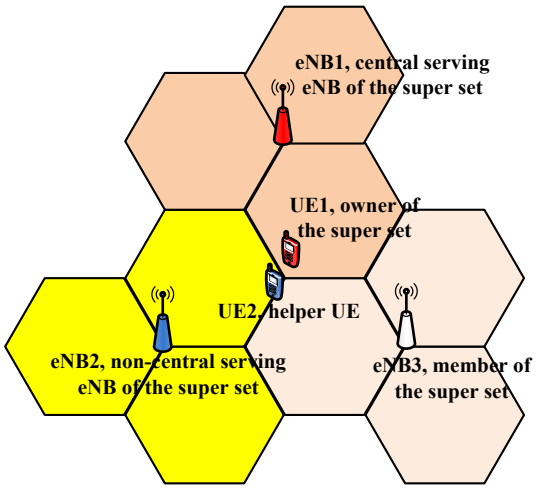


Figure 5.2: Super set formation based on Figure 1 example

It should be noted that in the above case, and similar cases, there are two levels of coordination among the eNBs. The first one is based on the conventional concept of the coordination sets. For example, eNB2 as the serving eNB of the UE2 coordination set collaborates with non-serving eNBs of the UE2 coordination set (eNB1, and eNB3) to enhance UE2 performance. Based on the UUC algorithm, eNB2 has complete information on the pieces that UE2 uploads to its coordination set (which is constructed by eNB2, eNB1 and eNB3). The second level of coordination comprises being members of the super set. The eNB2, as a non-central serving eNB of the UE1 (i.e. the owner UE)

super set is responsible to send control information about the uploaded pieces by UE2 (the helper UE) to the central serving eNB of the super set, which is eNB1 (i.e. the serving eNB of the owner UE). The important point here is that the responsibilities of the eNBs at the first level are independent from the second level. The eNB2, regardless of the fact it is part of superset or not, takes care of the upload of UE2 based on the UUC mechanism.

5.4.2 The UEs with Partially Same Subset of the eNBs as Their Coordination Set

In this section, some of the eNBs in the coordination sets of the both UEs are similar and some of them are not. In other words, there is at least one eNB in one of the coordination sets, which is not a member of the other coordination sets. In addition, there is at least one common eNB between the coordination sets. Based on the above, the following scenarios can occur regarding the coordination sets of the owner UE and its helper UE.

5.4.2.1 Same Serving eNBs and Different non-Serving eNBs

This case corresponds to row 7 of Table 5.1, where both coordination sets have the same serving eNB and different sets of non-serving eNBs. Therefore, in this case, a Type I super set is sufficient to address the UEs upload. Based on the previous sections, in this scenario, there is only one central serving eNB and there is no non-central serving eNBs. In other words, the super set consists of a central serving eNB and one or more eNB members. For example, in Figure 5.3, eNB1 is the serving eNB of both UE1 and UE2 coordination sets. The eNB2 and eNB3 are the non-serving eNBs of the UE1 coordination set and eNB3 and eNB4 are the non-serving eNBs of the UE2 coordination set. As defined earlier, the set of non-serving eNBs of two UEs coordination set are not same when there is at least one eNB in one of the sets which is not a member of the other set (such as eNB4 in this example). Finally, in this example, eNB1 has the upload information of both coordination sets since it is the serving eNB of both of them.

5.4.2.2 Different Serving eNBs and same non-Serving eNBs

This scenario corresponds to row 6 of Table 5.1. Since the coordination sets of the UEs

(i.e., UE1 as the owner UE and UE2 as the helper UE) have different serving eNBs, the network requires that updates concerning the uploaded pieces of the helper UE be sent to the serving eNB of the owner UE. Therefore, a Type 2 super set is established to perform the second level of the coordination. For example, in Figure 5.4, eNB2 and eNB4 are the serving eNBs of UE1 and UE2 respectively. In addition, eNB1 and eNB3 are the non-serving eNBs of the both UEs. The central eNB of the super set is eNB2, and eNB4 is the non-central serving eNB of the super set. The eNB1 and eNB3 are the non-serving eNBs of the super set. As a result, eNB4 sends the updates dedicated to the uploaded piece of the helper UE to the eNB2.

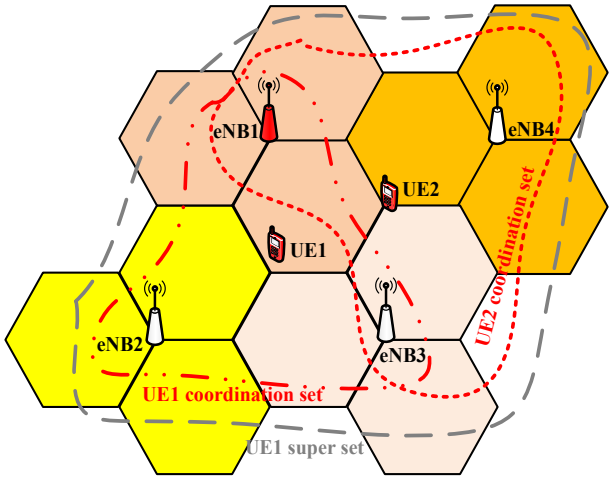


Figure 5.3: Two UEs with same serving eNB and different non-serving eNBs

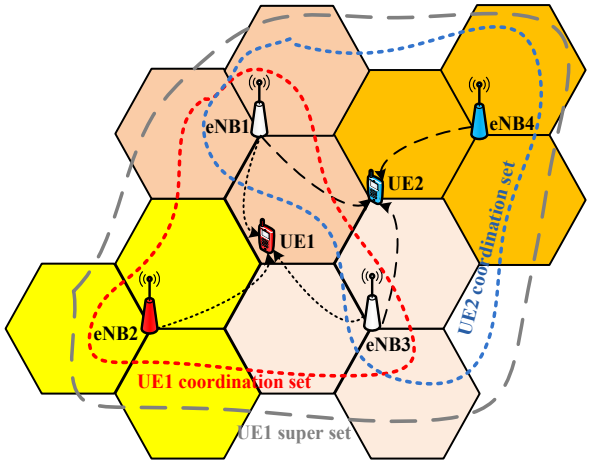


Figure 5.4: Coordination sets with different serving eNB and same non-serving eNBs

5.4.2.3 Different Serving eNB and non-Serving eNBs

This scenario corresponds to row 5 of Table 5.1. Figure 5.5 illustrates an example of this scenario, in which eNB1, eNB2 and eNB3 form the coordination set of UE1, and eNB3, eNB4 and eNB5 form the coordination set of UE2. Furthermore, eNB2 and eNB4 are the serving eNBs of UE1 and UE2 respectively. The pairs of (eNB1, eNB3) and (eNB3, eNB5) are the non-serving eNBs of UE1 and UE2 respectively. Since the coordination sets of the UEs (the owner UE and the helper UE) have different serving eNBs, a Type 2 super set is established to perform the second level of the coordination (as in the previous section). In this example, eNB2 is the central serving eNB of the super set and eNB4 is the non-central serving eNB of the super set. In addition, eNB1, eNB3 and eNB5 are the non-serving eNBs of the super set. When UE2 (the helper UE) uploads a piece to its coordination set successfully, eNB4 as the serving eNB of the UE2 and as the non-central serving eNB of the super set of UE1 (the owner UE), sends a message to the eNB2 (the central serving eNB of the super set) to notify eNB2 of the successful upload of that piece.

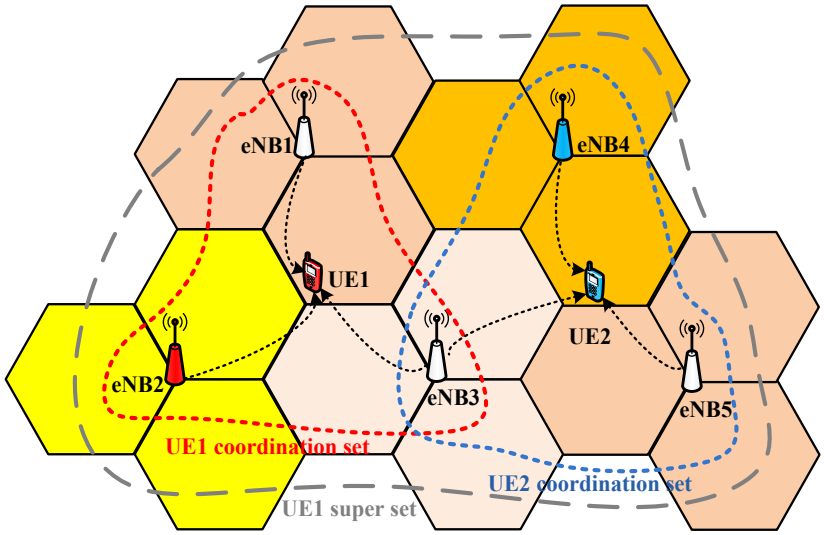


Figure 5.5: Coordination sets with different serving eNB and non-serving eNBs

5.4.3 UEs with Different Subsets of the eNBs as Their Coordination Set

In this category, the UEs (the owner UE and the helper UE) use different sets of eNBs as their coordination sets. The following scenario can occur for the owner UE and a helper UE.

5.4.3.1 Different Serving eNB and Different non-Serving eNBs

This scenario corresponds to the first row of Table 5.1, in which two coordination sets do not have any common eNBs in their sets. As a result, the serving eNB of the owner UE establishes a Type 2 super set (considering itself as the central serving eNB of the super set) to support the data file upload from multiple sources. For example, in Figure 5.6, UE1 uses eNB1, eNB2 and eNB3 as its coordination set, whereas UE2 uses eNB4 and eNB5 for the same purpose. The rest of the settings are similar to the previous sections. Considering Figure 5.6 as an example of this situation, eNB2 is the central serving eNB of the super set and eNB4 is the non-central serving eNB of the super set. In addition, eNB1, eNB3 and eNB5 are the non-serving eNBs of the super set.

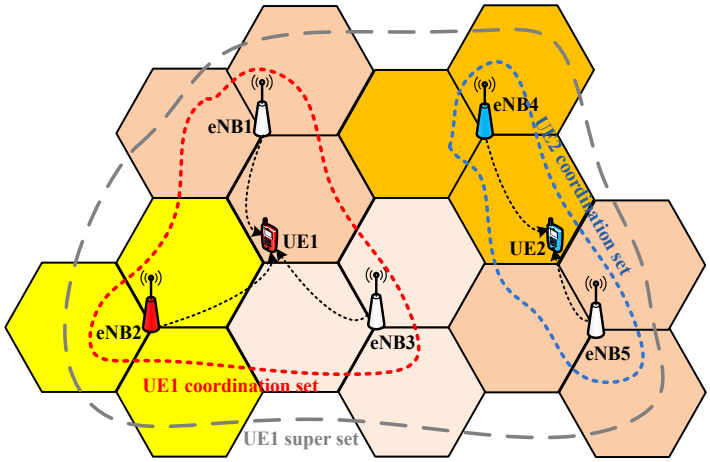


Figure 5.6: Coordination sets of two UEs with different subset of the eNBs.

5.5 Super Set Messages

The previous sections have defined the different scenarios that need to be supported by the UUC to support multiple coordination set cooperation for single data file upload. This section defines the changes to the current structure of the some of the messages of the UUC algorithm and defines some new messages as well. In the following sections, in all the message structures, the first field ('message id' field) illustrates the type of the message and assists the receiver to understand what the different fields of this message are and how it should deal with them. In addition, the second and the third fields of the messages structure are the destination (receiver) and the source (sender) of the messages

respectively. Although the name of the second and the third fields may be different in the various messages, their concept is same. Also, the last field of the messages structures ('transmission counter' field) shows that if the received messages is an initial transmission or a retransmission of an earlier message. In this case, value '1' means that this message is an initial transmission and value N, where N is an integer and $1 < N < 5$, shows that this message is a retransmission of a previously transmitted message. Clearly, the value of the last field of the messages structure shows the number of the transmissions of the same message.

5.5.1 Upload Request Message

The *Upload Request* message structure in the UUC is updated to the format in Table 5.2. The 'sender UE id' and 'serving eNB id' are the id of the UE that sends this upload request and its serving eNB respectively. If the 'owner UE' field and 'sender UE' field carry same value, it means that the owner UE is the one who has sent this message. Otherwise, the helper UE is the transmitter of this message. If the helper UE sends this upload request to the eNBs of its coordination set, the 'data file size' field represents amount of data that the helper UE wants to upload (which usually is smaller than the actual data file size).

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer	2	✓	✓	✓
2	serving eNB id	integer		✓	✓	✓
3	sender UE id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	string		✓	✓	✓
6	data file size	double (in MB)		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 5.2: The structure of the 'Upload Request' message

5.5.2 Helper UE Detected Message

In addition to the handshake response to the helper UE, there are further actions that may

be required to be taken by the serving eNBs based on the type of the UE. If the sender of the *Upload Request* message is a helper UE and the serving eNB of the helper UE is not as same as the serving eNB of the owner UE, then the serving eNB of the helper UE should notify the serving eNB of the owner UE of the helper UE upload status. The serving eNB of the helper UE uses the message structure in Table 5.3 to inform the serving eNB of the owner UE of the status of this helper UE. In Table 5.3, the 'receiver eNB id' shows the id of the serving eNB of the owner UE. The 'sender eNB id' field represents the id of the serving eNB of the helper UE, which is the actual sender of this message. In addition, the 'list of the non-serving eNBs id' field represents the id of the non-serving eNBs of the helper UE coordination set. The 'helper UE id' and 'owner UE id' fields represents the id of the helper UE and the owner UE respectively. The 'data file name' shows the name of the file that the owner UE and the helper UEs are trying to upload.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	receiver eNB id	integer		✓	✓	✓
3	sender eNB id	integer		✓	✓	✓
4	helper UE id	integer		✓	✓	✓
5	list of non-serving eNBs id	integer array		✓	✓	✓
6	owner UE id	integer		✓	✓	✓
7	data file name	string		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 5.3: The structure of the 'Helper UE Detected' message

5.5.3 Super Set Message

Upon receiving a *Helper UE Detected* message from other serving eNBs, the serving eNB of the owner UE creates a super set (considering itself as the central serving eNB of the super set) to support the upload of the data file. The central serving eNB sends a *Super Set* message (Table 5.4) to the serving eNB of the detected helper UE to inform it that it should start to send its information to the central serving eNB. In Table 5.4, the

"non-central serving eNB id" field represents the receiver eNB id and the 'central serving eNB id' has the id of the sender of this message. This message includes the id of the owner UE and the helper UE and their serving eNBs as well as the data file name. Finally, it allows the non-central eNB to forward the received pieces to the MME by writing 0 in the correspondence field. If this value is 1, the non-central serving eNB sends the received pieces to the central serving eNB. In addition, a value of 2 for this field means that the non-serving eNB should not send the received piece to any destination and it should retain the received piece. It is possible to extend the values that this field can take to support more other cases. It is trivial that in such a case we need to make sure that both the sender and receiver know about the meaning of the newly added value.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	non-central serving eNB id	integer		✓	✓	✓
3	central serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	helper UE id	integer		✓	✓	✓
6	data file name	string		✓	✓	✓
7	forward received pieces to MME	integer		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 5.4: The structure of the 'Super Set' message

5.5.4 Received Piece Report Message

The non-serving eNBs of the super set send their feedback regarding the received pieces to the central serving eNB by using the message structure described in Table 5.5. This table includes information about the received piece, the helper UE that uploaded this piece, the owner UE, and the serving eNBs of these UEs. This message helps the central serving eNB (the serving eNB of the owner UE) to keep track of the upload process and provide accurate information about the uploaded pieces for the owner UE. In Table 5.5, the 'central serving eNB id' field represents the receiver eNB id and the 'non-central serving eNB id' has the id of the sender of this message. Both the 'address of the first

byte of the received piece' field and 'address of the last byte of the received piece' field represent the exact address of the location of the piece in the actual data file.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	central serving eNB id	integer		✓	✓	✓
3	non-central serving eNB id	integer		✓	✓	✓
4	helper UE id	integer		✓	✓	✓
5	owner UE id	integer		✓	✓	✓
6	data file name	string		✓	✓	✓
7	piece number	integer		✓	✓	✓
8	piece size	integer		✓	✓	✓
9	address of the first byte of the received piece	integer		✓	✓	✓
10	address of the last byte of the received piece	integer		✓	✓	✓
11	transmission counter	integer		✓	✓	✓

Table 5.5: The structure of the 'Received Piece Report' message

5.5.5 Piece Acknowledgment Message

After receiving a report about a received piece (that was uploaded by a helper UE) at the non-central serving eNB coordination set, the central serving eNB of the super set uses the message structure, described in Table 5.6, to provide a *Piece Acknowledgement* message corresponding to the received piece for the owner UE. This message informs the owner UE that a helper UE could upload a piece of the data file successfully.

5.6 Type 1 Superset Flow

This type of the super sets, defines a method for UE's (with same serving eNB) employing the UUC [8] to dynamically and optimally select their super set based on the concept of a CoMP coordination set. This version is used when all the UEs that are participating in the upload process of a specific data file employing a same eNB as their serving eNB in their coordination set. In other words, the helper UE has same serving eNB as the owner UE. However, the helper UE and the owner UE use different set of

eNBs as their non-serving eNBs in their coordination set (like UE 1 and UE 2 in Figure 5.3). Table 5.7 shows the information of the coordination sets of the UE1 and UE2 according to Figure 5.3.

1	Field Name	Field Type	Value	UPS	VPS	DPS
2	message id	integer		✓	✓	✓
3	owner UE id	integer		✓	✓	✓
4	serving eNB id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	piece number	integer		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 5.6: The structure of the 'Piece Acknowledgement' message

	eNBs of the Coordination set	Serving eNB	Non-serving eNBs
UE1	eNB1, eNB2, eNB3	eNB1	eNB2, eNB3
UE2	eNB1, eNB3, eNB4	eNB1	eNB3, eNB4

Table 5.7: Coordination Set (CS) information for UE1 and UE2 from Figure 5.3

To do so, the helper UE needs to ask for the upload resources from its serving eNB (which in this scenario, this eNB is the serving eNB of the owner UE as well). Therefore, it sends the updated *Upload Request* message (Table 5.2) to the serving eNB. Upon receiving a message from the helper UE, the serving eNB creates a super set based on the coordination set of the both the owner UE and the helper UE (check 5.3 for super set definitions and rules). In our example from Table 5.7, the super set will look like Figure 5.3. The rest of the upload process is like [8]. After correct reception of each of the pieces, the serving eNB updates the DFDC and it determines the transporter eNB to send that piece to the MME. After uploading all the pieces, the owner UE sends *Done* message to the serving eNB. The serving eNB perform the final check and if there is any problem, it informs UE to resolve the problem. Finally, the MME tries to reconstruct the final based on the received pieces and the DFD. Figure 5.7 shows the type 1 super set creation.

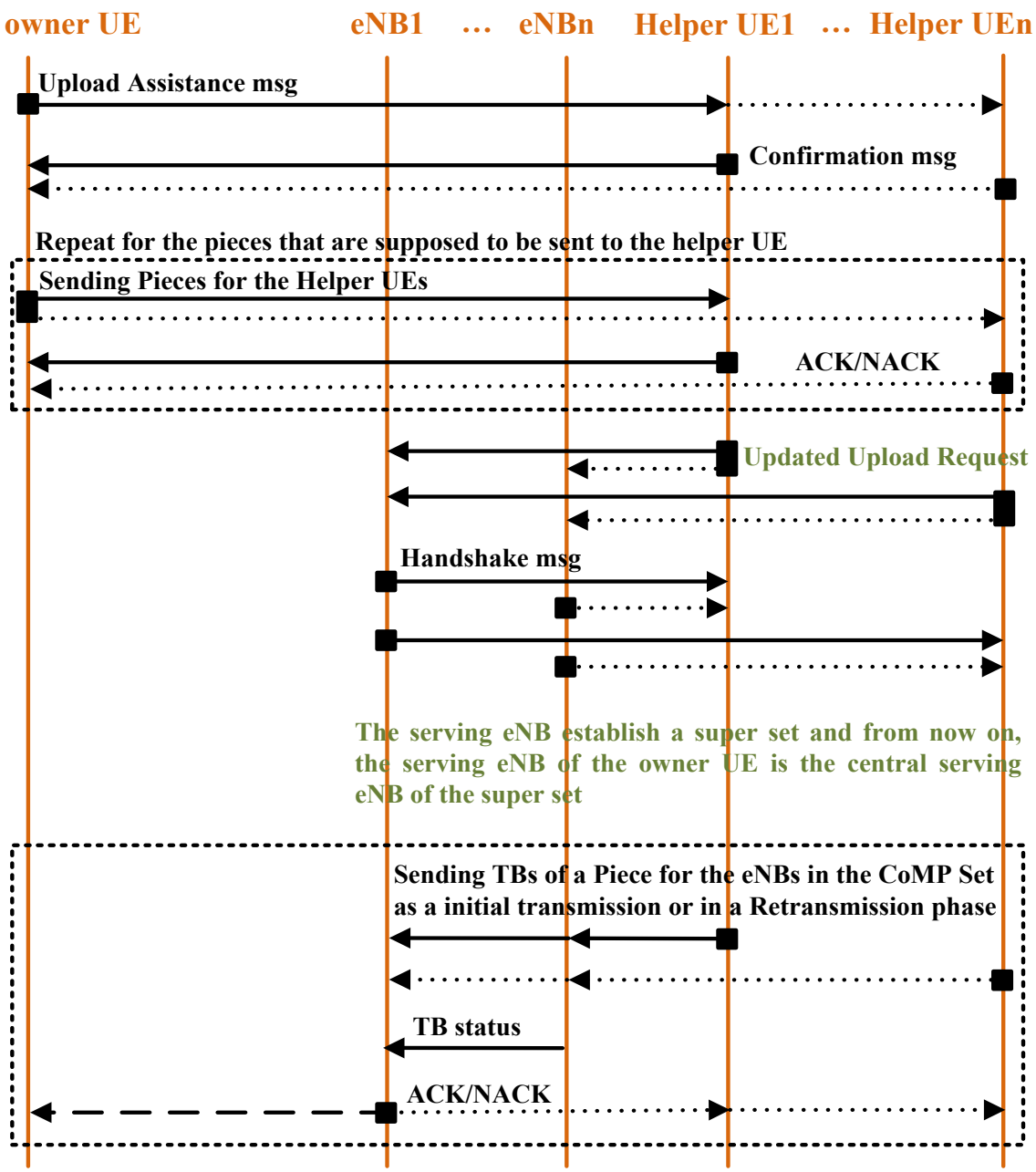


Figure 5.7: The helper UE pieces upload and the Type I super set creation (eNB1 is the serving eNB of the owner UE and helper UEs)

5.7 Type 2 Superset Flow

This type of the super sets, defines a method for UE's (without same serving eNB) employing the UUC [8] to dynamically and optimally select their super set based on the concept of a CoMP coordination set. This version is employed when the helper UE and

the owner UE use different eNBs as their serving eNB. This situation leads to creation of type II super set for supporting the UEs upload. The upload process starts as the one has been described in the previous section. However, when the helper UE sends the updated 'Upload Request' to its serving eNB, that serving eNB sends a *Helper UE Detected* message to the serving eNB of the owner UE. Upon receiving this message, the serving eNB of the owner UE establish a type II super set and announce itself as the central serving eNB of the super set. The central serving eNB informs the serving eNB of the helper UE about the super set information by using the *Super Set* message. From now on the serving eNB of the helper UEs are the non-central serving eNBs of the owner UE super set. When the non-central serving eNBs receive a piece from the helper UEs, they send a 'Received Piece Report' to the central serving eNB and the latter sends a *Piece Acknowledgement* message to the owner UE. This process let the owner UE to have a clear idea about the pieces that are already uploaded (Figure 5.8).

5.8 Owner UE Super Set with More Than One Helper UE

The formation of the super set when an owner UE has more than one helper UE is not any different from the scenarios when there is only one helper UE. The reason is simple: the serving eNB of the owner UE deals with each *Helper UE Detected* message independently and it does not matter how many helper UE will be detected by other eNBs.

In this section, an example of such a scenario is discussed. As shown in Figure 5.9, UE1 is an owner UE that with the aim of UE2 and UE3 (as the helper UEs) wants to upload its data file. UE1 coordination set includes eNB1, eNB2 and eNB3, where eNB2 is the serving eNB. UE2 coordination set consists of eNB3, eNB4 and eNB5, where eNB4 is the serving eNB. UE3 coordination set includes eNB6 and eNB7, where eNB6 is the serving eNB. When UE2 and UE3 send their upload requests to their serving eNBs, their serving eNBs (eNB4 and eNB6 respectively) send a *Helper UE Detected* message to the serving eNB of the owner UE, which is eNB2. Upon receiving these messages, eNB2 creates a super set with the following characteristics and it sends back *Super Set* message to the eNB4 and eNB6. The rest of this process is like the one in the previous sections.

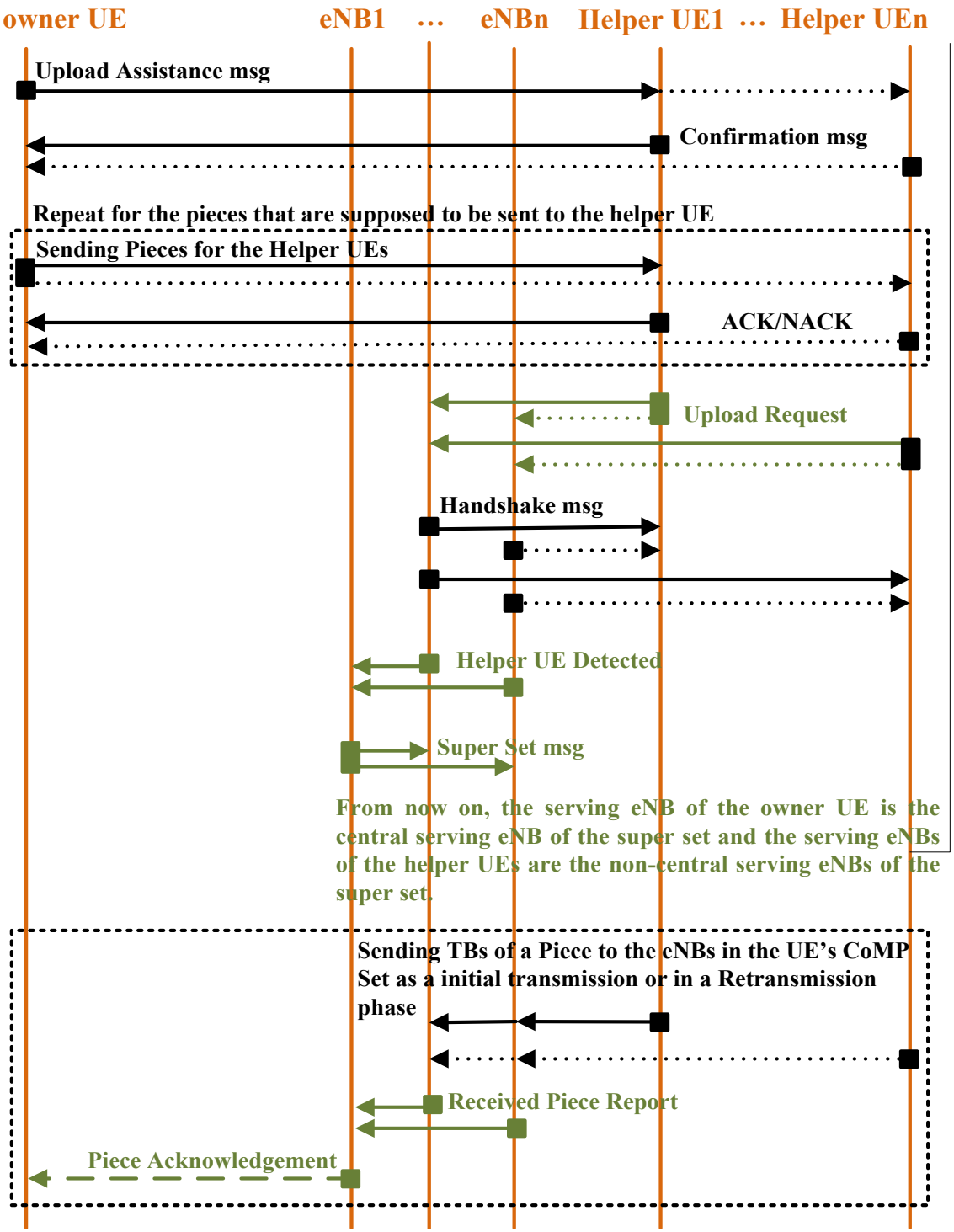
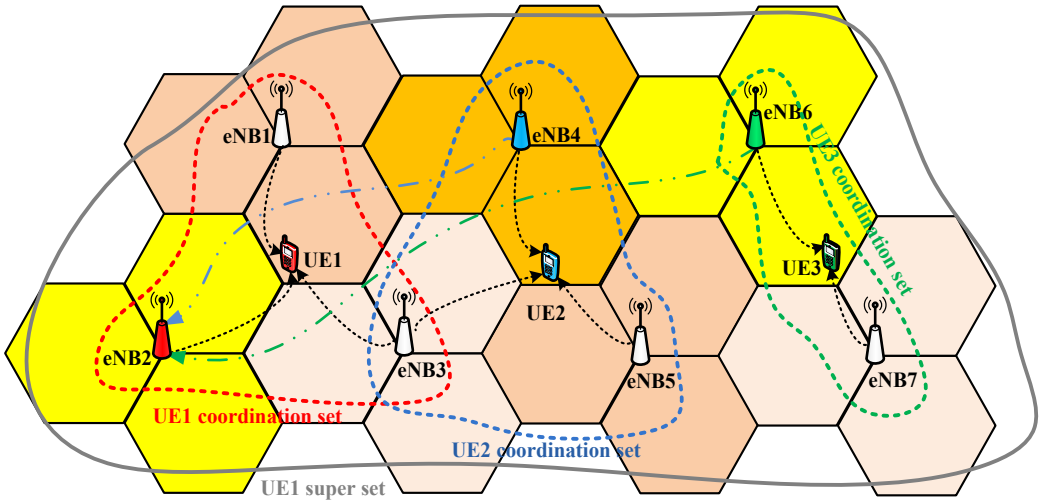


Figure 5.8: The helper UE pieces upload and the Type II super set creation (the owner UE and the helper UE have different serving eNBs)



UE1 super set information:
Owner UE: UE1
Helper UEs: UE2 and UE3
Central serving eNB: eNB2
Non-central serving eNB: eNB4 and eNB6
eNB members (non-serving eNBs): eNB1, eNB3, eNB5 and eNB7

Figure 5.9: Super set of an owner UE with two helper UEs

5.9 Summary

In this chapter, we presented the Super Set concept as a solution for the fixed coordination set limitation. This extension improves the UUC in such a way that the UEs that are involved in the upload process of an owner UE can use different coordination set of the eNBs. We divided super sets into two categories based on the serving eNB of the coordination sets of the owner UE and the helper UE. In addition, we discussed how we could employ each of these types in the UUC method. Moreover, we defined the required messages for the implementation of the Super Set concept as well as the structure of those messages. Finally, it should be noted that the other upload methods and protocols could use this method to extend their functionality as well. This method can be used in the download process as well.

6 Handover

LTE-Advanced uses different techniques such as Coordinated Multi-Point (CoMP) to provide consistent services for the users' equipment (UEs) regardless of their location in the coverage area. CoMP refers to a set of base stations (eNBs) that are coordinated jointly and dynamically. With the implementation of CoMP, eNBs can support joint scheduling of transmissions and provide joint processing of the received signals in order to improve system performance. Specifically, CoMP eNBs form coordination sets for which the main objective is to manage interference to enhance the performance of UEs especially for the cell edge users. However, providing suitable consistent services for the USs is challenging, especially when some of the UEs are close to the cell borders. This group of users suffers from two problems: the higher interference from the neighboring cells, and the long distance between them and their serving Base Station (eNB), which is located at the cell center. The longer distance between sender and receiver leads to weaker received signal power at the receiver side. Both interference and reduced signal power result in poor mobile network services for the cell edge UEs.

In UUC, we discussed a new method to enhance the cell edge users upload in the LTE-Advanced networks. This method uses multiple UEs upload power to speed up the upload process of a specific UE. These UEs are served by same sub set of the eNBs as their coordination set and they do not change their serving eNB or their host cell during the upload process. In [9], the authors elaborated a method to show how the fixed coordination set limitation in the UUC can be removed. The proposed method in [9] improves the UUC in such a way that the UEs that are involved in the upload process of an owner UE can use different coordination set of the eNBs. However, in the both of [8] and [9], there is an assumption that the UEs do not change their serving eNB or their host cell. This assumption needs to be removed because in the real world the users can be mobile and possibly change their position in the network one or more times over the duration of the UUC upload. As a result, the operators should be able to address this kind

of situation and provide consistence services for their users. In this chapter, we are going to show how we can extend the UUC to support handover for the UEs that need to change their serving eNB in order to maintain a communications link of sufficient quality. In addition, in a separate section, we discuss the handover concept when we are dealing with super sets in the mobile network. To do so, we discuss the new steps and messages that we need to add to the LTE handover process [29] to adapt it for the UUC algorithm.

6.1 The Handover Algorithm

The LTE standard defines the handover process. However, when a user employs the UUC [8] for the upload process, more tasks require to be done for a successful handover operation. The current serving eNB of the owner UE should forward the latest information of the owner UE upload process to the new serving eNB. Beside this, the current serving eNB should inform the other eNBs that are involved in the UE upload/download process about the handover process. This thesis provides detailed steps and required messages structure for these purposes. In addition, if the owner UE and the helper UE upload process is supported by a super set, and the owner UE wants to change its serving eNB, beside the said tasks, the current serving eNB should forward the super set information to the new serving eNB as well. If the helper UE is the one that wants to change its serving eNB, then the current serving eNB of the helper UE should pass the super set information to the new serving eNB of the helper UE and it should inform the central serving eNB (serving eNB of the owner UE) about these changes.

The main purpose of this section is to show how we could extend the UUC to let it support the handover process. In the LTE standard [34] the serving eNBs of the UEs set the measurement thresholds according to the area restriction. Considering these thresholds, the UEs send their measurement report to their serving eNB. The serving eNB of the UE uses this information to determine if the signal strength of the neighbor eNBs is better than the signal strength of the current serving eNB of the UE. At this step, let us assume that there is an eNB with this situation. Therefore, the current serving eNB of the UE selects that eNB to be the new serving eNB of the UE. Therefore, the current serving

eNB sends a *Handover Request* message to the selected eNB. This message includes the necessary information to prepare the selected eNB for the handover process. Upon receiving the *Handover Request*, the selected eNB performs the admission control to see if there are available resources to accept this session. If the selected eNB wants to grant the resources then it sends *Handover Request Acknowledge* message to the current serving eNB. After that, the current serving eNB, the selected eNBs, and the UE perform some more steps to avoid data loss during the handover process (such as an X2 bearer establishment between the eNBs to carry the user data during the handover). These steps provide the basis for handover in the LTE protocol. Figure 6.1 tells how the handover process works in the LTE (the concepts in this figure borrowed from [34]).

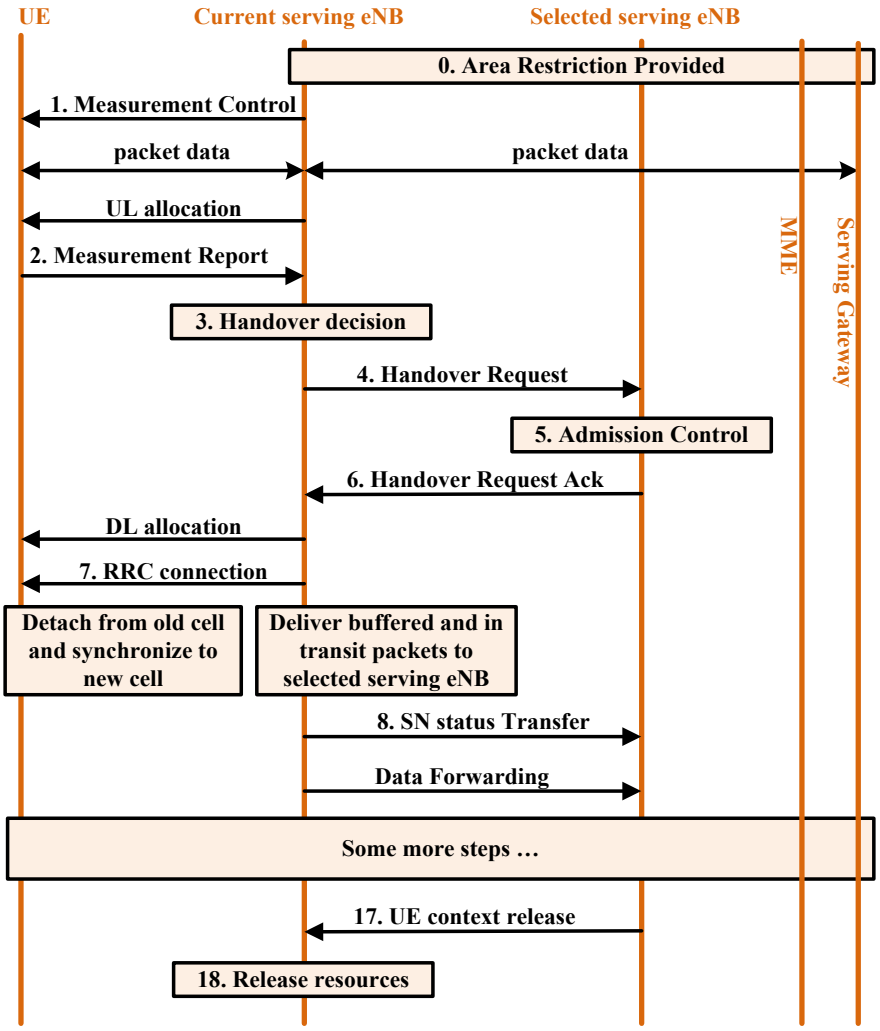


Figure 6.1: Basic handover process in LTE [34]

6.2 UUC Handover Messages and Structures

We continue this thesis with the new steps and messages that we need to add to the LTE handover process to adapt it for the UUC algorithm. Before that, it worth saying that in the following sections, in all the message structures, the first field (i.e. 'message id' field) shows the type of the message and helps the receiver to understand what are the different fields of this message and how it should deal with them. In addition, the second and the third fields of the messages are the destination (receiver) and the source (sender) of the messages respectively. Although the name of the second and the third fields may be different in the various messages but the concept is same. Also, the last field of the messages structures ('transmission counter' field) shows that if the received messages is an initial transmission or a retransmission of an earlier message. In this case, value '1' means that this message is an initial transmission and value N, where N is an integer and $1 < N < 5$, shows that this message is a retransmission of a previously transmitted message. Clearly, the value of the last field of the messages structure shows the number of the transmissions of the same message. In the following, we discuss the detail of required messages and their structure.

6.2.1 Updating the Selected Serving eNB about the Owner UE Upload Status

In the UUC algorithm, the serving eNBs of the owner UEs keep track of the upload process of the owner UEs by saving the related information in a corresponding *Data File Descriptor Complement* structure. Upon selecting a new serving eNB for the owner UE (based on the previous section), the current serving eNB should send both the *Data File Descriptor* and the latest version of the *Handover DFDC* message (Table 1) to the selected (new) serving eNB. This means that the selected serving eNB is updated by the latest status of the UE upload/download process.

In Table 6.1, the 'new serving eNB id' and 'current serving eNB id' fields represent the id of the new serving eNB and current serving eNB of the owner UE in order. The 'owner UE id' shows the id of the owner UE of the data file. The 'DFD type' field shows which one of the UUC methods (UPS, VPS or DPS) was used by the owner UE to divide the data file into the pieces. The 'data file name' and 'data file size' fields show the name and

the size of the file that the owner UE wants to upload. The 'piece length' field shows the size of the pieces (this field is being used when the owner UE employs UPS or VPS). The 'number of the pieces' field shows how many different pieces should be uploaded to complete the upload of the data file and the 'number of the received pieces' field tells how many of them already got uploaded by the UEs. The 'list of the received pieces' field includes the id of the received pieces. The corresponding address of these received pieces is saved in the 'address of the first byte of the received pieces' and 'address of the last byte of the received pieces' fields.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	new serving eNB id	integer		✓	✓	✓
3	current serving eNB	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	DFD type	string		✓	✓	✓
6	data file name	string		✓	✓	✓
7	data file size	float (in MB)		✓	✓	✓
8	piece length	float array		✓	✓	
9	number of the pieces	integer		✓	✓	
10	number of the received pieces	integer		✓	✓	✓
11	list of received pieces	integer array				
12	address of the first byte of the received pieces	integer array		✓	✓	
13	address of the last byte of the received pieces	integer array		✓	✓	
14	retransmission	boolean		✓	✓	✓
15	transmission counter	integer		✓	✓	✓

Table 6.1: The structure of the 'Handover DFDC' message

It should be noted that during the handover process, after DFDC submission to the new serving eNB, the current serving eNB might receive information about the owner UE upload process. This missing information in the DFDC message may cause a retransmission request by the new serving eNB (because it does not know about this

information). Therefore, the next serving eNB suspends the retransmission process until it receives another message (to be described later) from the current serving eNB (which will be the previous serving eNB by that time) that indicates the new serving eNB should remove the retransmission request suspension. This message is sent to the new serving eNB to make it sure that from now on the retransmission request for a missing piece is not a redundant process. Besides that, upon receiving the *Handover DFDC* message, the new serving eNB will set a timer and after a configurable period, it will automatically remove that suspension as well (if it has not received a message from the previous serving eNB to cancel the request suspension).

6.2.2 Notify the Other eNBs about the Changes and Receive Their Response

Besides sending the updates to the selected serving eNB, the current serving eNB must also inform the other eNBs that are involved in the UE upload/download process about the handover changes. These eNBs are part of the UEs coordination sets. The current serving eNB uses the *Serving eNB Changed* message (Table 6.2) to update other non-serving eNBs regarding the serving eNB change. The 'receiver eNB id' represents the id of the destination eNB. The 'next serving eNB id' and the 'owner UE id' fields show the id of the next serving eNB of the owner UE and the id of owner UE respectively. The rest of the fields are as already discussed in previous sections.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	receiver eNB id	integer		✓	✓	✓
3	current serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	next serving eNB id	integer		✓	✓	✓
6	retransmission	boolean		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 6.2: The structure of the 'Serving eNB Changed' message

It worth saying that the previous serving eNB of the owner UE accepts the messages

about the owner UE upload process until it receives a *eNB Changes Confirmed* message from all the eNBs that it has sent them the *Serving eNB Changed* message or until end of a specific time interval. Table 6.3 shows the structure of the *eNB Changes Confirmed* message. The 'previous serving eNB id' field in Table 6.3 as the same value as the 'current serving eNB id' field in Table 6.2. In addition, 'sender eNB id' shows the id of the sender of this message, which is equal to the value of the 'receiver eNB id' field in Table 6.2.The rest of the fields are as already discussed in the previous sections.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	previous serving eNB id	integer		✓	✓	✓
3	sender eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	next serving eNB id	integer		✓	✓	✓
6	retransmission	boolean		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 6.3: The structure of the 'eNB Changes Confirmed' message

6.2.3 Last Update from Previous Serving eNB of the Owner UE

As we said, the previous serving eNB of the owner UE may receive packets regarding the upload process of the owner UE even after it has been replaced by another eNB. In this situation, the previous serving eNB forwards all the received information to the current serving eNB of the owner UE by using the message structure in Table 6.4 (*New Update* message). This process continues until the previous serving eNB receives the *eNB Changes Confirmed* messages from all the eNBs that it sent a *Serving eNB changed* message or after a specific time interval. After that, the previous serving eNB discards the received packets that are related to the upload process of this owner UE. In Table 6.4, the 'piece holder eNB id' field shows which one the eNBs has the complete uploaded piece. Later on, the new serving eNB can ask the piece holder eNB to forward that piece to new destination (i.e. such as the MME or the new serving eNB).

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	new serving eNB id	integer		✓	✓	✓
3	current serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	received piece id	integer		✓	✓	✓
7	piece holder eNB id	integer		✓	✓	✓
8	retransmission	boolean		✓	✓	✓
9	transmission counter	integer		✓	✓	✓

Table 6.4: The structure of the 'New Update' message

In addition, after receiving all the *eNB Changes Confirmed* messages or after a specific time interval, the previous serving eNB sends the *Last Update* message (Table 6.5) to the new serving eNB. By receiving this message, the new serving eNB removes suspension on the retransmission request process.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	serving eNB id	integer		✓	✓	✓
3	previous serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	retransmission	boolean		✓	✓	✓
7	transmission counter	integer		✓	✓	✓

Table 6.5: The structure of the 'Last Update' message

6.3 Sequence Diagram of the UUC Handover Process

As we said earlier, we employed the basic handover process from LTE as the base of our work. Subsequently we have added steps to the handover process making it compatible with the UUC as well. Figure 6.2 shows the sequence diagram of handover process in the

UUC algorithm. This figure illustrates the details that we discussed in the previous sections. In addition, it clearly shows the differences between this extended versions of the handover process with the one that already exist in the LTE protocol (follow the dash line).

As seen in Figure 6.2 (after step 8), the current serving eNB sends the required information about both the owner UE and its upload status to the new serving eNB. The latter gets ready for the rest of the owner UE upload, and it suspends the retransmission process for a specific time. The current serving eNB is also responsible to inform the other non-serving eNBs of the coordination set of any changes. Once it receives the ACK message from the non-serving eNBs, it sends the last update message as well as any newly received data to the new serving eNB. Finally, the new serving eNB removes the retransmission process suspension, and it assists the owner UE with the rest of the upload process.

6.4 UUC Handover Process with Super Sets

In the UUC algorithm, the authors assumed that a helper UE and its helper UEs use same subset of the eNBs as their coordination set. The super set concept introduced in the super set report removes this limitation [9]. The main goal of developing the super set concept was to let the owner UE and its helper UEs to use the same or a completely different subset of eNBs partially as their coordination sets, as well as using the same subset of the eNBs as their coordination set. Therefore, in the extended version of the UUC the UEs that are trying to upload different parts of a same data file (such as an owner UE and its helper UEs) are eligible to communicate with different subsets of the eNBs as their coordination set. This extended version of the UUC can be improved to support the handover process as well. In the previous chapter, we investigated the usage of coordination sets and super sets in the different situations that can occur when a helper UE wants to cooperate with an owner UE in the upload process. In this context six valid cases were defined that should be considered. These six modes can be categorized into three main groups as follows (A, B, C):

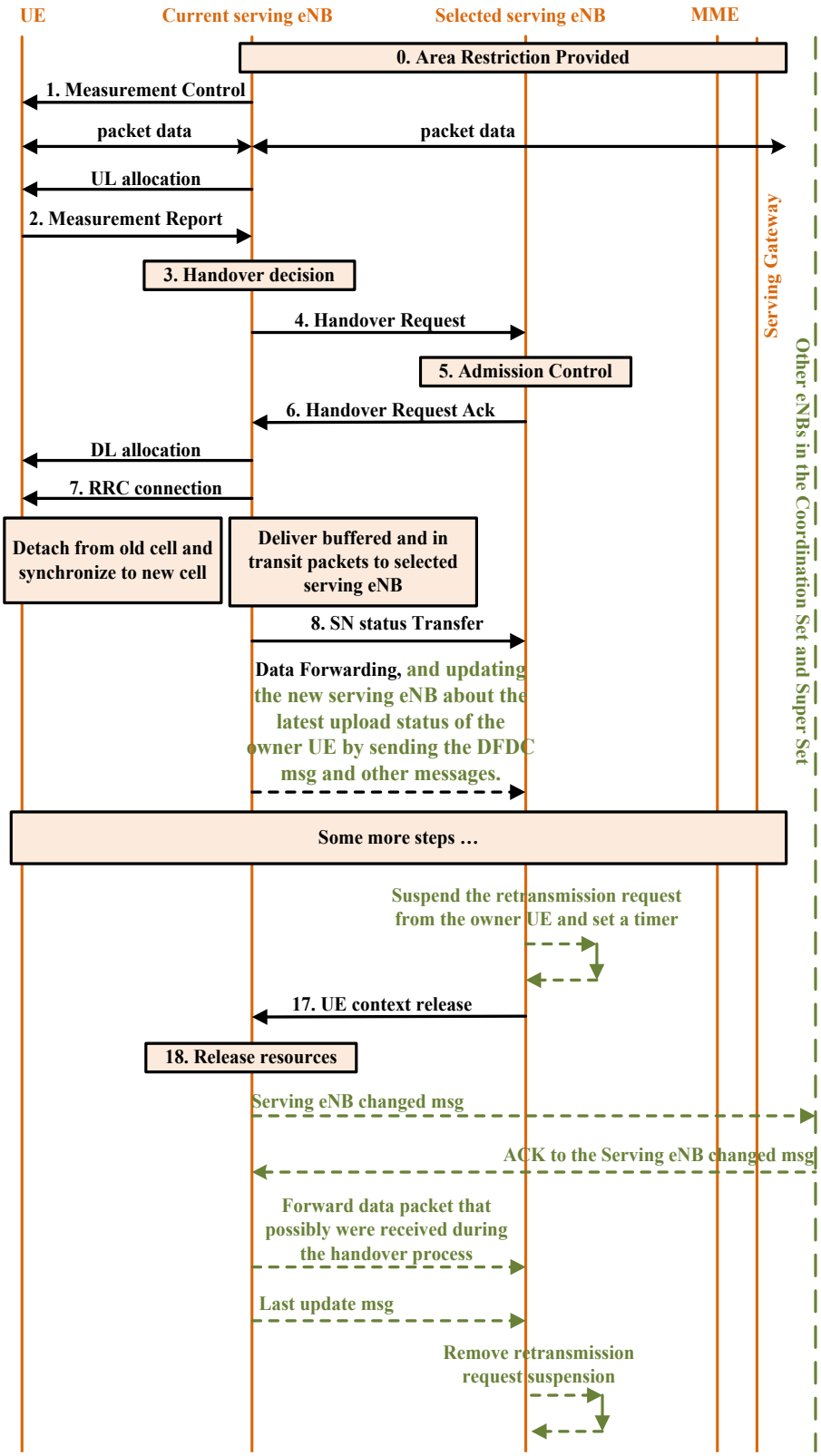


Figure 6.2: Sequence diagram of handover process in the UUC algorithm

A. The UEs use the same subset of the eNBs as their coordination set

1. They have the same serving eNB and same non-serving eNBs.
2. They have a different serving eNB and different non-serving eNBs.

B. The UEs use partially same subset of the eNBs as their coordination set

3. They have same serving eNBs and different non-serving eNBs.
4. They have different serving eNBs and same non-serving eNBs.
5. They have a different serving eNB and different non-serving eNBs.

C. The UEs use different subset of the eNBs as their coordination set

6. They have a different serving eNB and different non-serving eNBs.

The first mode (A1) does not require a super set and single coordination set is good enough to handle the upload process of multiple UEs. In the rest of five modes (from A2 to C6), we use the notion of a super set to handle multiple coordination set cooperation. Now let us assume there are an owner UE and its helper UE that are trying to upload different pieces of a same data file. The possible coordination set or super set structure that they are using is one of the above said modes. Now, let us assume that during the upload/download process a UE wants to change it location. This movement can lead to one of the following events: losing a connection with an eNB, establishing a connection with a new eNB and changing the serving eNB with a non-serving eNB and vice versa. As a result, the UE movement can lead to changing the current mode of the coordination set or the super set structure into another mode and at the same time, it may lead to a handover process. Table 6.6 shows how a UE movement causes the super set structure to change from one mode to another one. This table only shows those mode changes that are possible with a single UE movement. In the rest of this section, we investigate how these modes are convertible to each other.

Let us say three issues before we continue with the rest of this section. Firstly, from now on, when we say a UE movement, we mean either the owner UE movement or the helper UE movement. Therefore, this phrase is valid for the both categories of UEs movement.

Secondly, we are using some examples to discuss all the possible types of the super sets mode changes, but one can define different examples for the same types of changes. Finally, the UE movement can have an effect on the super set structure. Based on the type of this effect, different tasks are needed. Table 6.7 shows how we need to deal with different types of changes in the super sets configuration.

	A1	A2	B3	B4	B5	C6
A1		✓	✓			
A2	✓				✓	
B3	✓				✓	
B4					✓	
B5		✓	✓	✓		✓
C6					✓	

Table 6.6: Mode changes between Super Set structures

In the following, we are going to study the different situations in which the super set configuration changes. Based on the type of these changes, the algorithm can determine which one of the said actions in Table 6.7 is required. Clearly to handle each of these actions, we need to define the proper steps and the required messages structure for each of these steps. These issues are investigated in the subsequent sections.

6.4.1 Mode Change Between (A1, A2) and (A1, B3)

A UE movement can change the owner UE and its helper UE coordination set (mode A1) to a super set with mode A2 and vice versa. Figure 6.3 shows how this conversion can occur. UE2 movement changes the configuration of its coordination set. These changes convert coordination set with mode A1 into the super set with modeA2. As seen in this figure, both of the UEs use same subset of the eNBs as their coordination set. Now let us assume that the initial state of the UEs and the eNBs is like the right side of the Figure 6.3 (mode A2). Then UE2 movement to the new position (like its position at left side of the Figure 6.3) change its coordination set structure (to mode A1). This mode changes is a bidirectional relation in the rest of the mode changes as well.

Type	Field	
I	UE movement effect	Changing the central serving eNB (C-S-eNB) of the super set
	Required action	Handover at Super Set level
	Description	The current C-S-eNB (which is the current serving eNB of the owner UE) is going to change. Therefore, the current C-S-eNB needs to perform the following steps: 1. It receives some information about the new central serving eNB of the super set (from the owner UE). 2. It sends the information of the super set to the new C-S-eNB. 3. It sends the information of the received pieces to the new C-S-eNB. 4. It updates other the eNBs about the changing C-S-eNB event.
II	UE movement effect	Changing of one of the non-central serving eNBs (non-C-S-eNB) of the super set
	Required action	Handover at Coordination Set level
	Description	A non-C-S-eNB of the super set (which is the current serving eNB of the coordination set of one of the helper UEs) is going to change, then the current serving eNB of the coordination set of the helper UE needs to perform the following steps: 1. It receives some information about the new serving eNB of the coordination set (from the helper UE). 2. It sends the information of the coordination set to the new serving eNB of the coordination set of the helper UE. 3. It sends the information of the received pieces to the new serving eNB of the coordination set of the helper UE.
III	UE movement effect	Changing of one of the non-serving eNBs of one of the UEs
	Required action	An update
	Description	The non-serving eNB is a member of the coordination set of a UE. The serving eNB of that coordination set needs to perform the following steps: 1. It receives the information about this event from the UE. 2. It informs the central serving eNB of the super set about this change.

Table 6.7: UEs movement effects on the configuration of the super sets

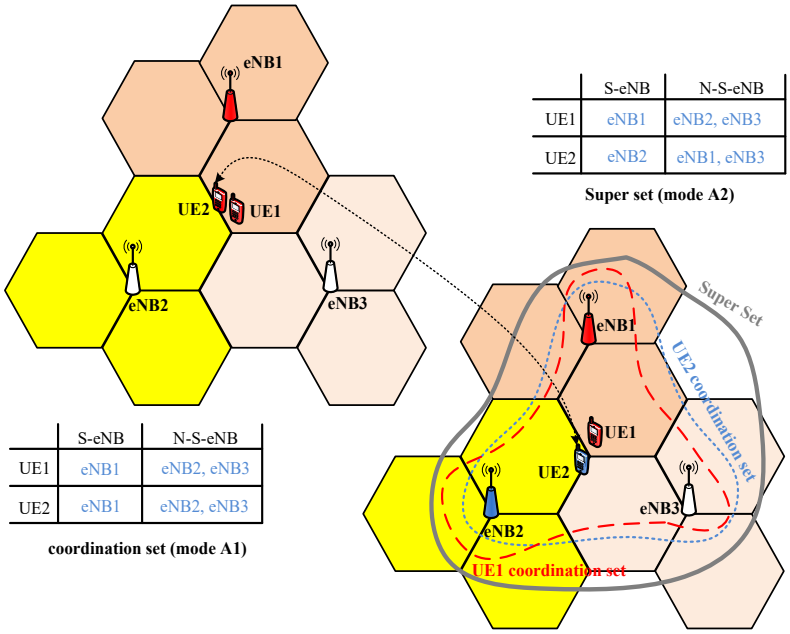


Figure 6.3: Coordination set and super set conversion (modes A1 and A2)

A UE movement can change the owner UE and its helper UE coordination set (mode A1) to a super set with mode B3 and vice versa as well. Figure 6.4 illustrates how UE2 movement changes the configuration of its coordination set and these changes convert coordination set with mode A1 into the super set with mode B3. As seen in this figure, both of the UEs use same subset of the eNBs as their coordination set.

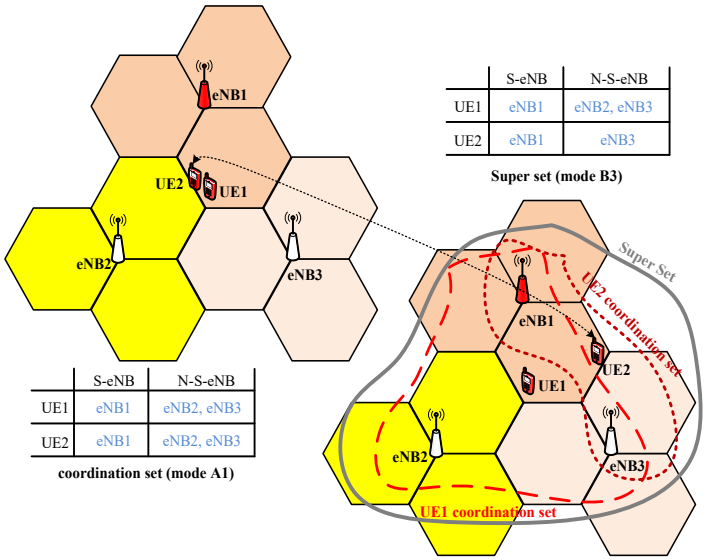


Figure 6.4: Coordination set and super set conversion (modes A1 and B3)

6.4.2 Mode Change Between (A2, B5)

A super set in mode A2 can convert to a super set with mode B5 and vice versa. In the left side of Figure 6.5, both UEs use same set of eNBs as their coordination set. Now, let us assume UE2 wants to change its position to the one depicted in the right hand side of the Figure 6.5. This movement disconnects the communication channel between UE2 and the eNB1. As a result, it causes a change in the UE2 coordination set (eNB1 is not part of the coordination set of the UE2 anymore) and it changes the super set structure from mode A2 to mode B3. On the other hand, let us assume that the initial position of the UE2 is as illustrated on the right hand side of the Figure 6.5 and it wants to move to a new position as illustrated on the left hand side of the same figure. Establishing a connection between UE2 and eNB1 changes the coordination set of the UE2 and finally the super set.

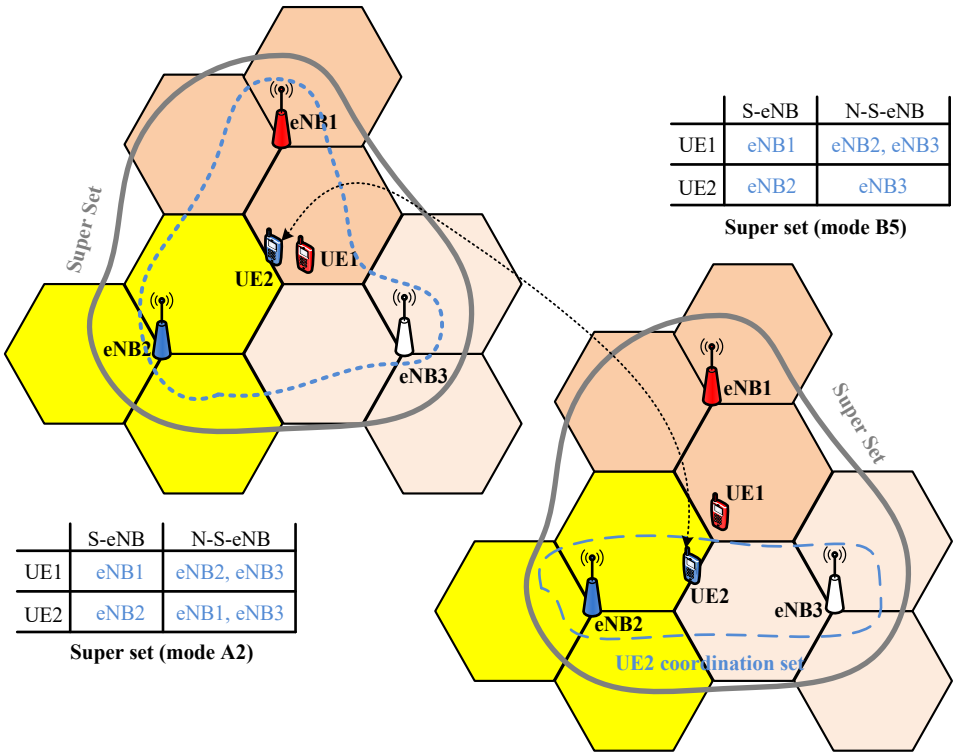


Figure 6.5: Super sets conversion (modes A2 and B5)

6.4.3 Mode Change Between (B3, B5)

An UE movement can also change the mode of super set from mode B3 to mode B5.

Figure 6.6 shows this event, where UE2 movement leads to poor communication channel between UE2 and eNB1 and a better communication channel between UE2 and eNB3. As a result, UE2 changes its serving eNB from eNB1 to eNB3. The latter changes the configuration of the super set as well. This conversion can occur from mode B5 (as the initial mode) to mode B3 as well.

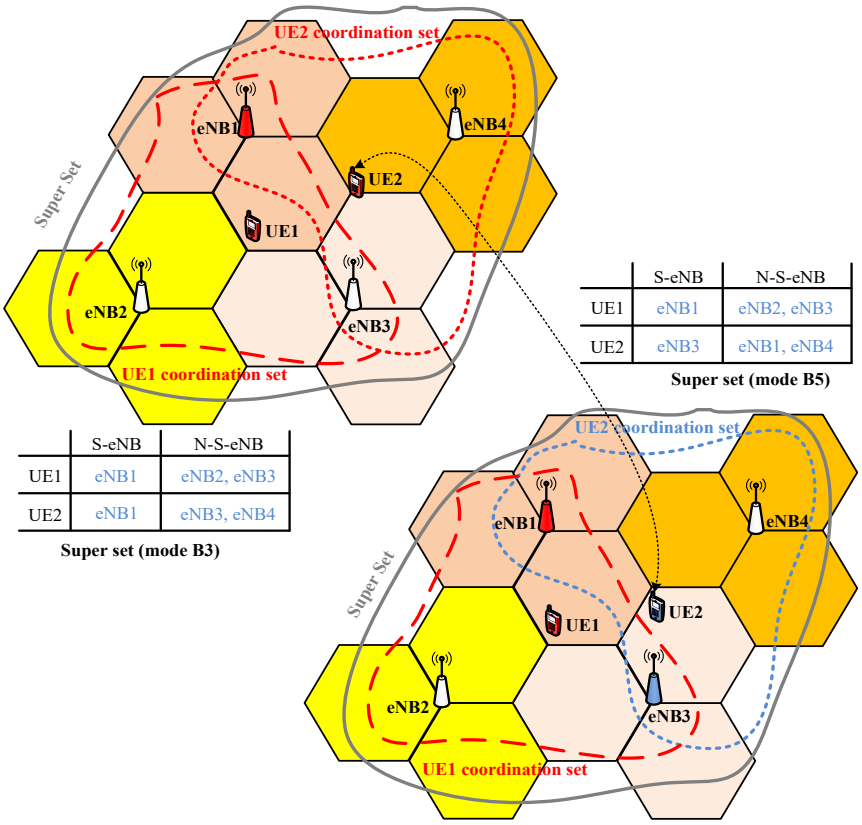
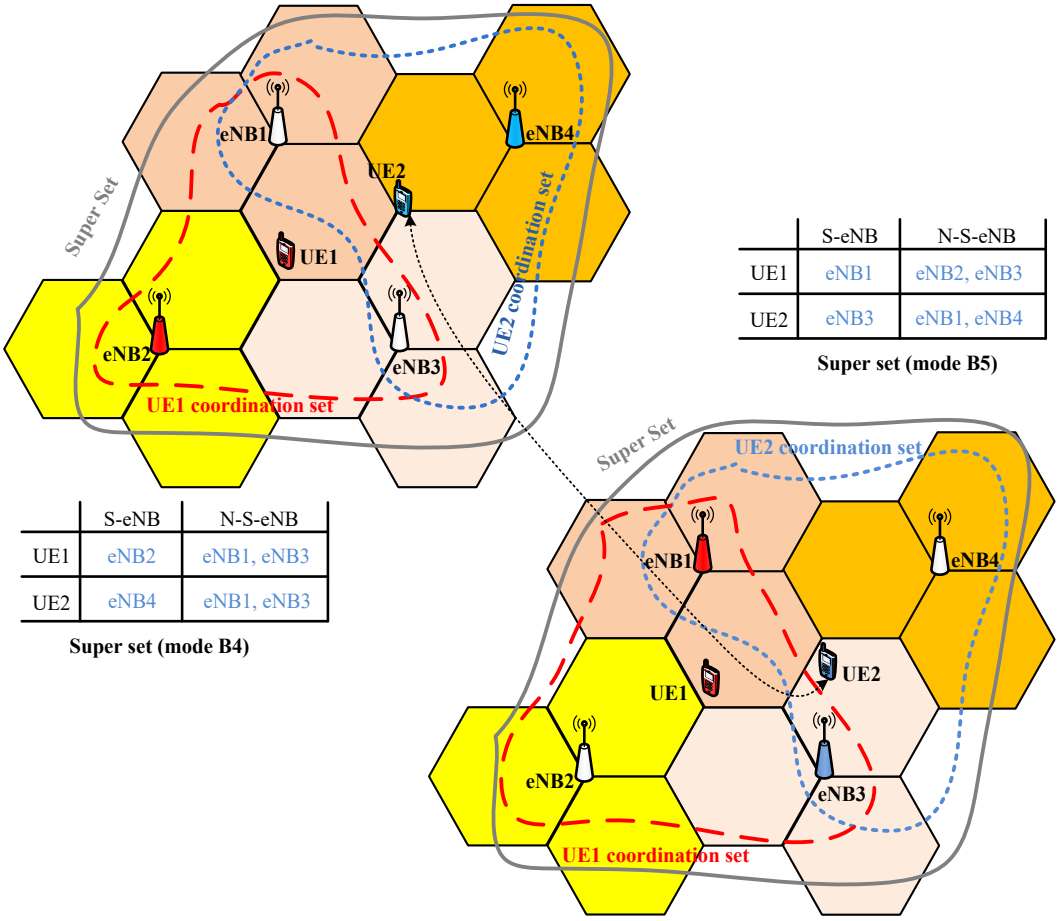


Figure 6.6: Super sets conversion (modes B3 and B5)

6.4.4 Mode Change Between (B4, B5)

Figure 6.7 illustrates that how a super set in mode B4 can convert to a super set with mode B5 and vice versa. In the top left side of this figure, UE2 uses eNB4 as its serving eNB. Then assume it decides to move toward eNB3. After a while, it understands that its communication channel status with eNB3 is better than the one with eNB4. Therefore, it starts the process of the changing the serving eNB. As a result, the configuration of the super set from mode B4 change to the mode B5. Again, same story is valid if we assume that the initial location of the UEs is like the one in the bottom right part of the Figure 6.7

and UE2 determines to go toward eNB4.



	S-eNB	N-S-eNB
UE1	eNB2	eNB1, eNB3
UE2	eNB4	eNB1, eNB3

Super set (mode B4)

	S-eNB	N-S-eNB
UE1	eNB1	eNB2, eNB3
UE2	eNB3	eNB1, eNB4

Super set (mode B5)

Figure 6.7: Super sets conversion (modes B4 and B5)

6.4.5 Formation Changes Between (B5, C6)

A super set in mode B5 can convert to a super set with mode C6 and vice versa. In Figure 6.8, UE2 movement terminates its communication channel with eNB3 and as a result, eNB3 is being removed from the UE2 coordination set. This event completely separates the coordination sets of two UEs from each other (bottom side of Figure 6.8) and changes the configuration of the super set. In addition, if we assume that the initial condition of the UEs is like the bottom part of the Figure 6.8, then UE2 movement toward eNB3 can add eNB3 to the coordination set of the UE2 (top side of Figure 6.8). In the following, we define the required messages to handle handover while we are using the extended UUC algorithm.

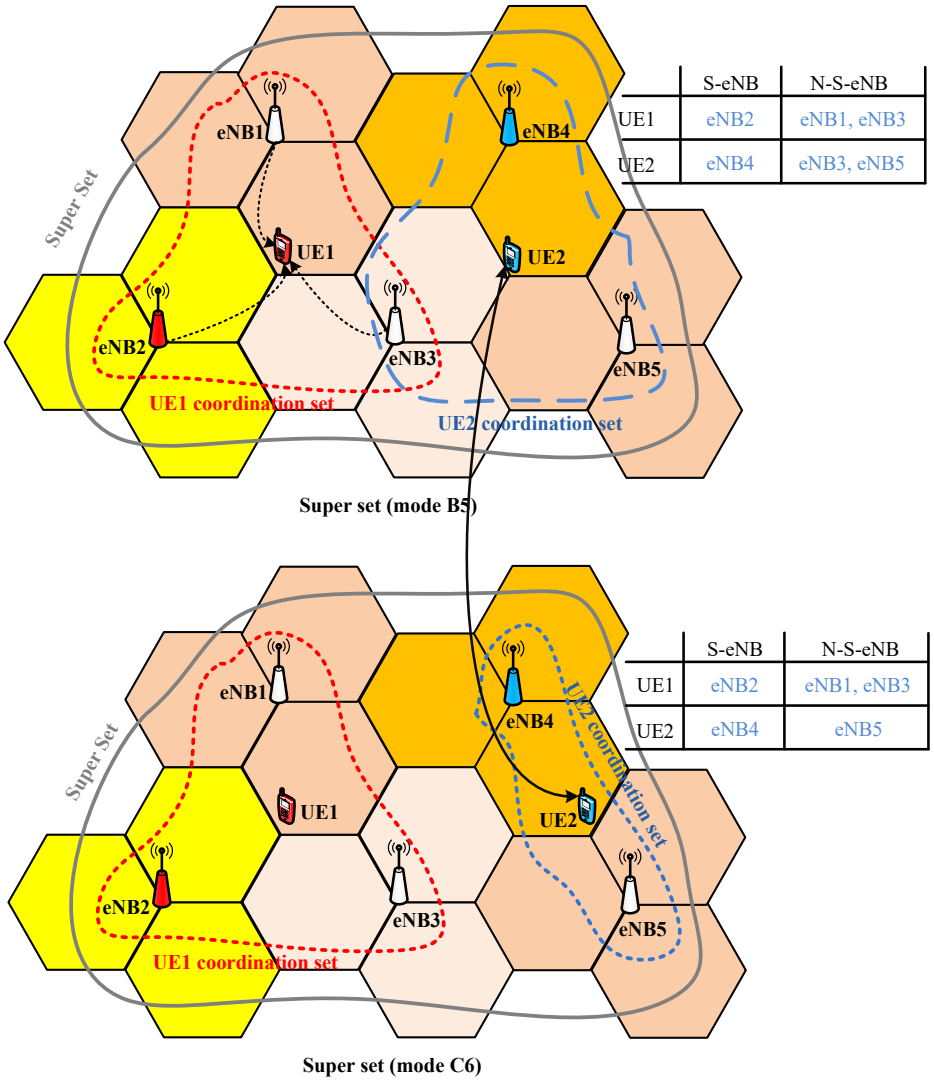


Figure 6.8: Super sets conversion (modes B5 and C6)

6.4.6 Messages to Support Handover over Super Sets

We described the required actions that need to be taken to update the new serving eNB of the UE while we are using the UUC in previous sections. Besides that, if we want to use the extended UUC (that uses super sets); the following actions should take place as well. If an owner UE wants to change its serving eNB (which in this example is the central serving eNB of the super set), it should send the information of the super set (including coordination sets and their members) to the new serving eNB (new central eNB of the super set). Table 6.8 shows the message structure that the current serving eNB uses to send the super set information to the next serving eNB. The 'coordination set' field is a

list of list pointers in which each pointer points to a list of eNBs ids. The first id belongs to the serving eNB of the coordination set and the rest of the list represents the non-serving eNBs of the same coordination set. If a helper UE wants to change its serving eNB, then the current serving eNB of the helper UE should pass the super set information to the new serving eNB (Table 6.9) and it should inform the central serving eNB (serving eNB of the owner UE) of these changes as well (Table 6.10). Figure 6.9 and Figure 6.10 show the sequence diagram of the UUC handover while it uses super sets and either the owner UE or the helper UE changes their serving eNB.

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	new central serving eNB id	integer		✓	✓	✓
3	current central serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	coordination sets info	list of list pointers		✓	✓	✓
7	retransmission	boolean		✓	✓	✓
	transmission counter	integer		✓	✓	✓

Table 6.8: The structure of the 'Super Set Info' message

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	new serving eNB id	integer		✓	✓	✓
3	current serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	central serving eNB	integer		✓	✓	✓
7	retransmission	boolean		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 6.9: The structure of the 'Pass Super Set Info' message

	Field Name	Field Type	Value	UPS	VPS	DPS
1	message id	integer		✓	✓	✓
2	central serving eNB id	integer		✓	✓	✓
3	current serving eNB id	integer		✓	✓	✓
4	owner UE id	integer		✓	✓	✓
5	data file name	integer		✓	✓	✓
6	new serving eNB id	integer		✓	✓	✓
7	retransmission	boolean		✓	✓	✓
8	transmission counter	integer		✓	✓	✓

Table 6.10: The structure of the 'Update Super Set Central eNB' message

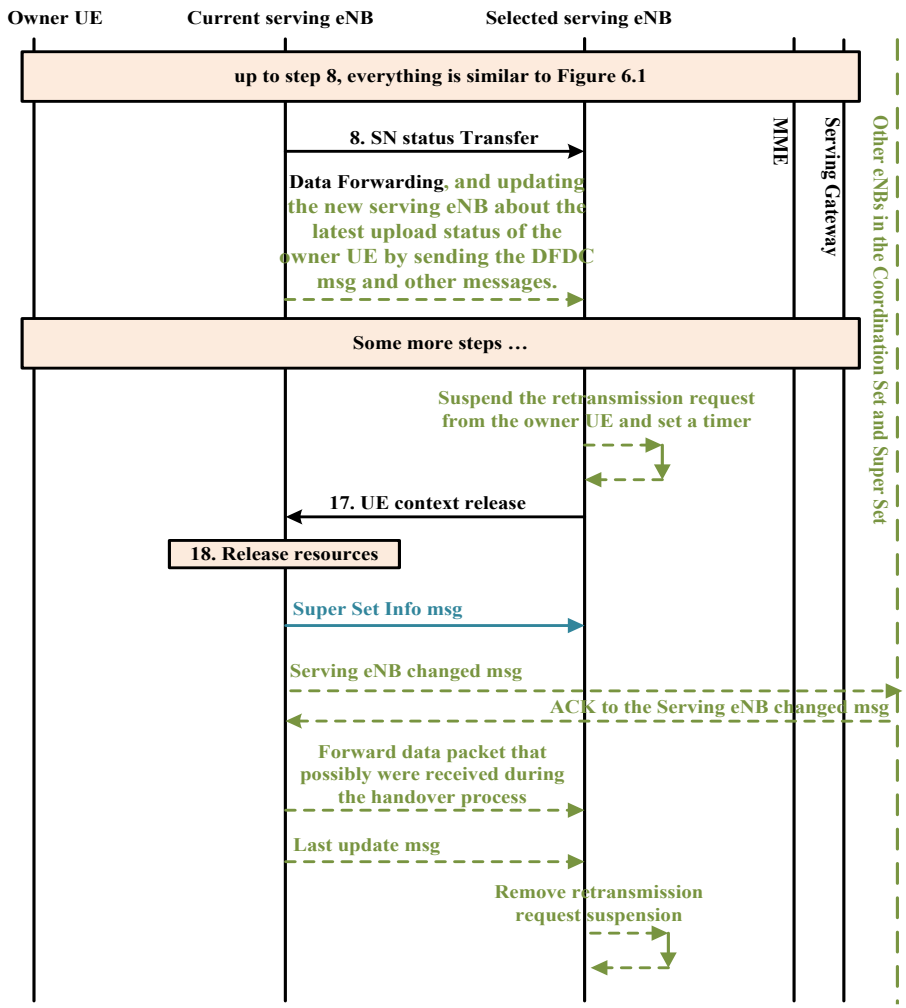


Figure 6.9: Sequence diagram of the UUC Handover (Owner UE uses Super Set)

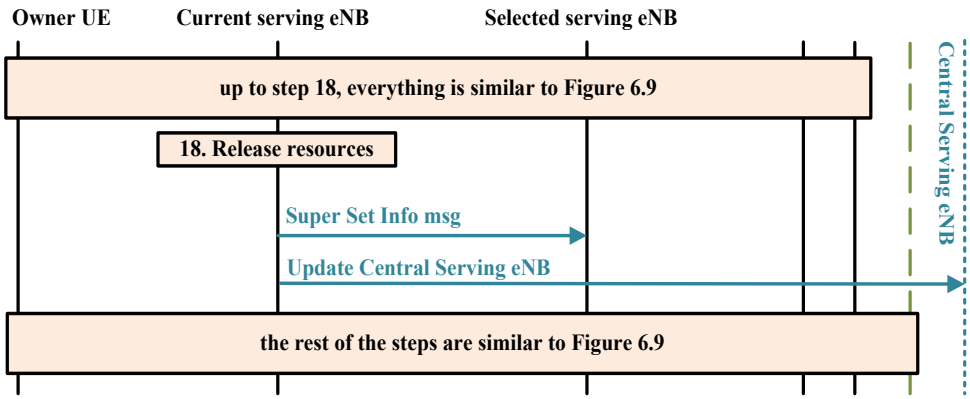


Figure 6.10: Sequence diagram of the UUC Handover (Helper UE in Super Set)

6.5 Summary

In this chapter, we presented a method to extend the UUC algorithm. This method allows UUC to support handover for the UEs that need to change their serving eNB in order to maintain a communications link of sufficient quality. In addition, we discussed the handover concept when we are dealing with super sets in the mobile network. The proposed method allows the UEs to remove one or more eNBs from their coordination set and add other eNBs into their coordination set in order to optimize the UE throughput.

7 Retransmission

LTE-Advanced uses Hybrid ARQ with soft combining as a retransmission scheme. Hybrid ARQ with soft combining is categorized into Chase Combining (CC) and Incremental Redundancy (IR). In IR, each retransmission sends a new redundancy version – i.e. a new set of coded bits different from the first (Figure 7.1). Chase combining resends the same bits, which can be soft combined with the first received TB (Figure 7.1). Both of these methods work as follows: the receiver asks for a retransmission up to three times if it cannot recover the data from the initial transmission. In addition, the receiver uses previous erroneous TBs for data recovery in addition to the new received TB. This method increases the accumulated received E_b/N_0 for each retransmission and increases the chances of recovering the original data from the received TBs. If after completing four transmissions, the eNBs cannot recover original data from the TBs, the MAC layer does not do anything about this TB and it leaves the error recovery for the upper layer (RLC layer).

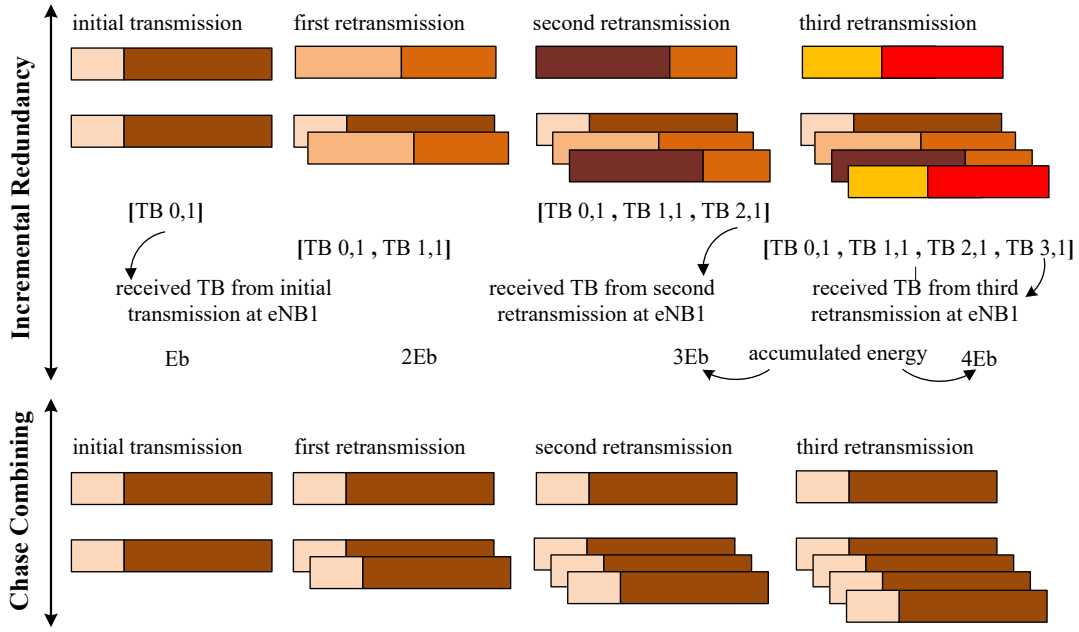


Figure 7.1: Hybrid ARQ with soft combining - Incremental Redundancy

If we consider the LTE-Advanced retransmission policy for a cell-edge UE, we can improve the way it works to save the cell-edge UE upload resources and time. We explain the problem by using an example in Figure 7.2. Let us assume there is a cell-edge UE with three eNBs in its range. These eNBs form a coordination set to support the UE upload process. Now, imagine the UE transmits a TB to the eNBs of its coordination set, and none of the eNBs of the coordination set can recover data from it. Therefore, the UE requires retransmitting this TB. As seen in Figure 7.2.A, the eNBs of the coordination set (eNB1, eNB2 and eNB3) receive the initial transmission of a TB. We assumed the received copy of the TB by all eNBs has an error and none of the eNBs can recover the data from the received copy at their side. Therefore, the serving eNB (eNB1) asks for the retransmission of the erroneous received TB from the sender UE. This UE retransmits the TB for the first time. In this situation, each of the eNBs of the coordination set received the second copy of the TB (Figure 7.2.B). Again, if none of the eNBs of the coordination set can recover the data from the different copies of the received TB at their side, the serving eNB asks for another retransmission, and the UE retransmits another copy of the TB (Figure 7.2.C). In case of unsuccessful data recovery from different copies of the received TB, this process can be repeated one more time (Figure 7.2.D).

The problem with these methods is that each transmission and retransmission consumes UE process time. For example, the initial transmission and the first two retransmissions provide an opportunity for the eNBs to perform chase combining based on the three TBs that are identical (Figure 7.2.C). At the same time, this means that three time slots are wasted to transmit the same TB. In addition, if we allow the eNBs to send retransmission requests for the UE in an independent manner, then another problem arises. Consider the case in which one of the eNBs receives the transmitted TB error free and the other two eNBs receive it with error. The second and third eNBs send retransmission requests to the UE directly. As a result, the UE starts the retransmission process. This process is wasteful since, the first eNB received the TB without error and in essence, a retransmission is not required. Rather, the eNBs could communicate and let each other know about the status of TB reception at their side. Our proposed method aims to reduce unnecessary retransmissions and enhance the UE's performance.

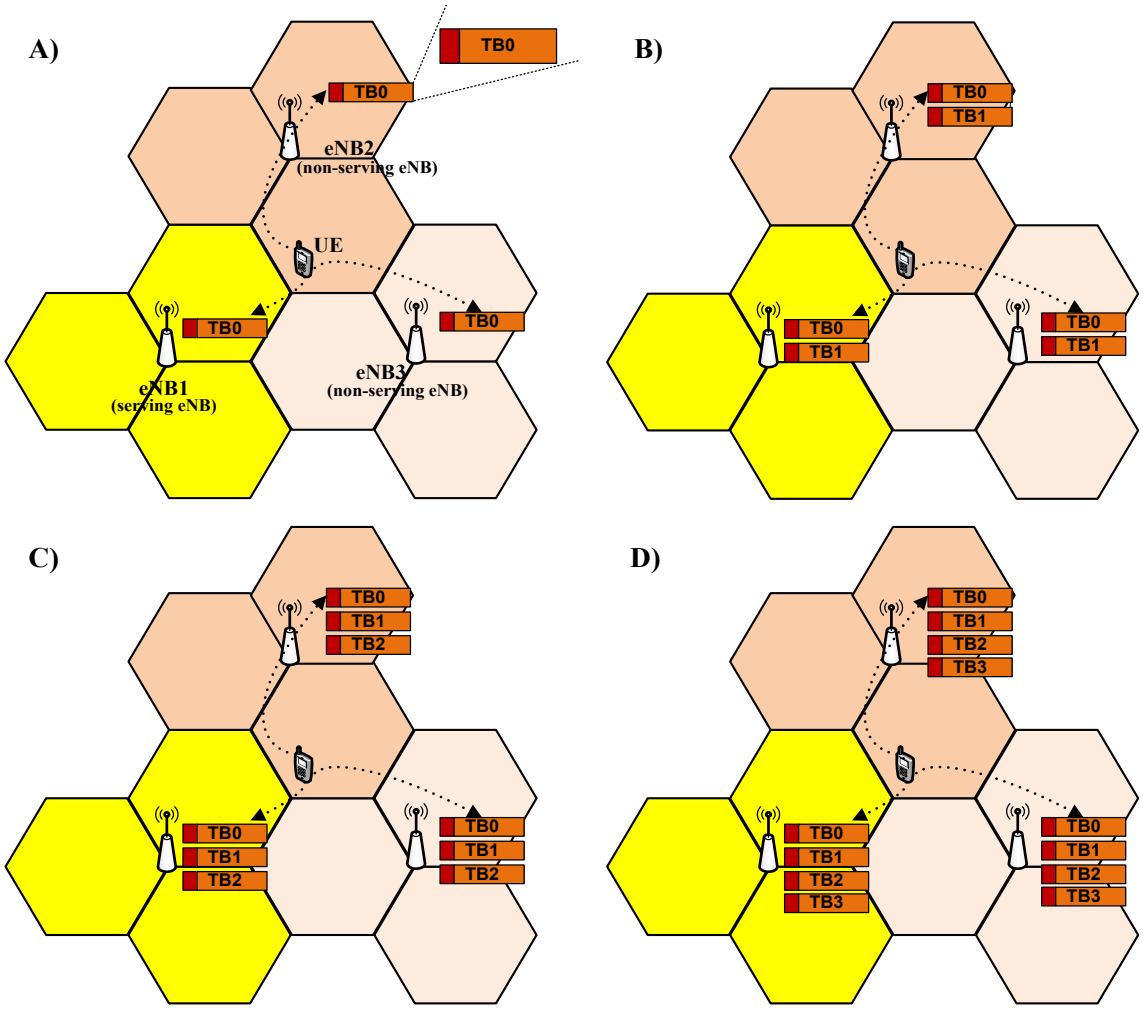


Figure 7.2: Chase Combining (mobile network perspective)

7.1 Retransmission Mechanism for the MAC Layer

Let us assume we have a coordination set of three eNBs that support a cell-edge UE upload, where eNB1 is the serving eNB, and eNB2 and eNB3 are the non-serving eNBs of the coordination set (similar to Figure 7.2.A). In this section, we discuss how the proposed retransmission mechanism works. The UE starts to upload the initial TB and each of the eNBs receives this TB. As shown in Figure 7.3, the serving eNB tries to recover the data from the received TB. Let us assume that this process is unsuccessful. Therefore, the serving eNB sends a control message for the non-serving eNBs of the coordination set to ask them about the status of this TB. In response to this message, the

non-serving eNBs send feedback about the received TB to the serving eNB. If at least one of them was able to receive the TB without error, the problem is solved, but if not, the serving eNB uses the *Forward the received TB* message to ask them to send their TB to the serving eNB. Note that during this time, the UE continues uploading other pieces. The non-serving eNBs use the *received TB* message to forward the received TB to the serving eNB. After receiving the TB from the other eNBs, the serving eNB is able to perform Chase combining on them as it is shown in Figure 7.4. As we said earlier, this process increases the chance of recovering the data from the TB's.

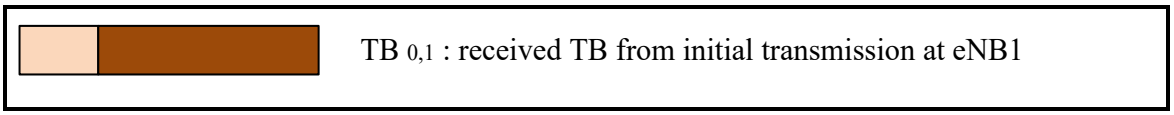


Figure 7.3: Received TB from initial transmission at eNB1 (the serving eNB)

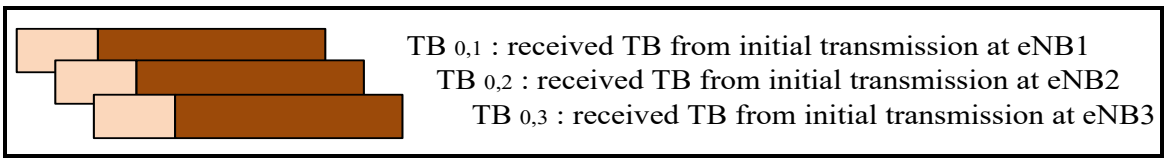


Figure 7.4: The serving eNB performs Chase Combining

If we compare this with the normal retransmission methods that we discussed, the UE does not need to repeat the upload two more times to provide three identical TBs for the eNBs to increase the chance of data recovery. Rather, the non-serving eNBs can send their TBs to the serving eNB. By doing this, there is the same amount of information at the serving eNB that can be used to recover the original data using Chase combining. If the first Chase Combining attempt does not work, the serving eNB sends a NACK message regarding this TB to the UE. The UE retransmits this TB based on incremental redundancy approach (i.e. using a different redundancy version than in the first transmissions). After receipt of the second redundancy version from the UE, the serving eNB tries to recover data using incremental redundancy based on the newly received TB and previously Chase-Combined TBs from the previous step (Figure 7.5). At the same time, the non-serving eNBs try to recover the TB data from the newly received

redundancy version (they just have the new TB and the previous one that they received in previous step).

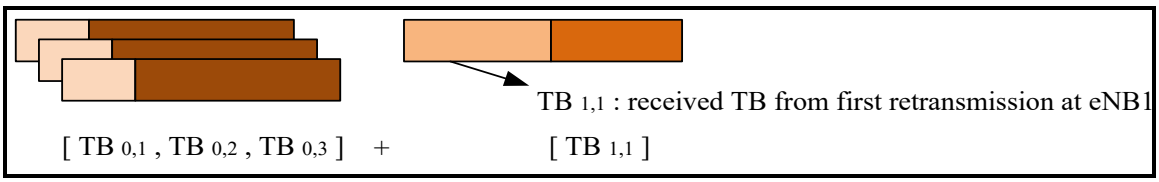


Figure 7.5: The serving eNB side after receiving first retransmitted TB

If none of the eNBs is able to recover the original data, the serving eNB asks the non-serving eNBs to send their TBs to the serving eNB. This means that the non-serving eNBs send their received copy of the TB ($TB_{1,2}$ and $TB_{1,3}$) to the serving eNB. After that, the serving eNB performs the chase combining based on the previous and new TBs as illustrated in Figure 7.6 to recover the original data.

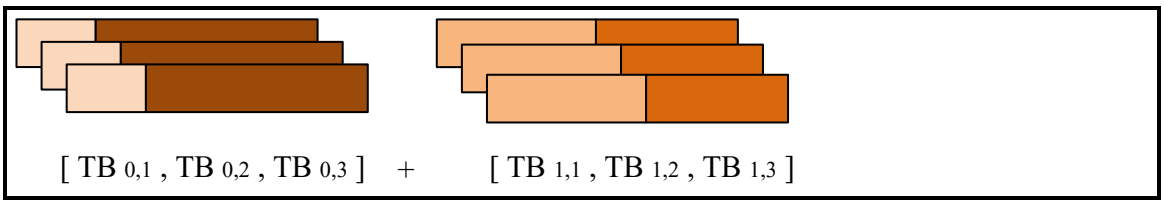


Figure 7.6: The serving eNB side after receiving other eNBs' 1st retransmitted TB

Again, if this process is successful, the serving eNB sends an ACK for the UE but if not, then it sends a NACK to the UE. In the latter case, the retransmission process continues up to three times. The next steps are shown in Figure 7.7, Figure 7.8, Figure 7.9 and Figure 7.10. In Figure 7.7, the serving eNB receives a new copy of the TB (from the second retransmission of the TB by the UE). If none of the eNBs of the coordination set can recover the data from the received copies of the TB, the non-serving eNBs send their latest received copy of the TB (the one from the second retransmission) to the serving eNB (Figure 7.8). After that, the serving eNB performs the chase combining based on the previous and new TBs. If the outcome of the data recovery is not successful, the serving eNB asks the UE to retransmit the TB for the third time.

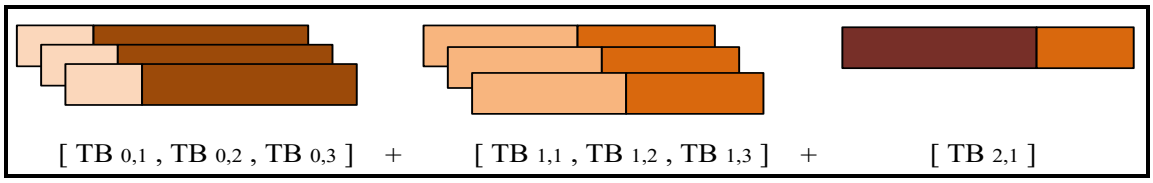


Figure 7.7: The serving eNB side after receiving second retransmitted TB

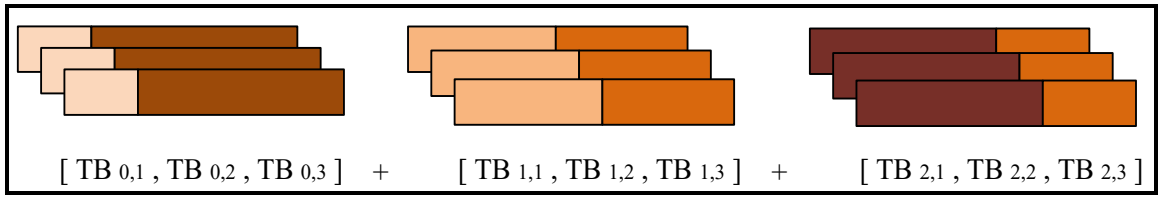


Figure 7.8: The serving eNB side after receiving other eNBs' 2nd retransmitted TB

In Figure 7.9, the serving eNB receives a new copy of the TB (from the third retransmission of the TB by the UE). Again, if none of the eNBs of the coordination set can recover the data from the received copies of the TB, the non-serving eNBs send their latest received copy of the TB (the one from the third retransmission) to the serving eNB (Figure 7.10). If after all these steps none of the eNBs are able to recover the original data from the received TBs, the MAC layer leaves the task of error free data recovery for the upper layer (RLC layer).

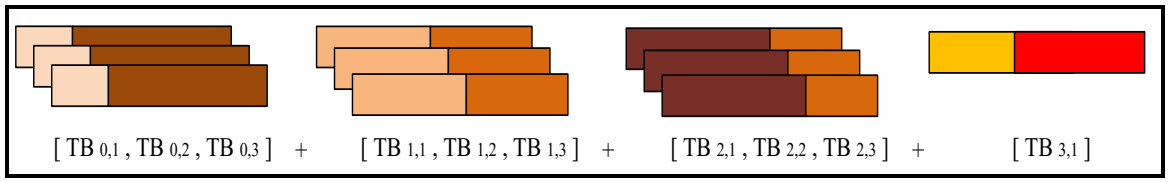


Figure 7.9: The serving eNB side after receiving third retransmitted TB

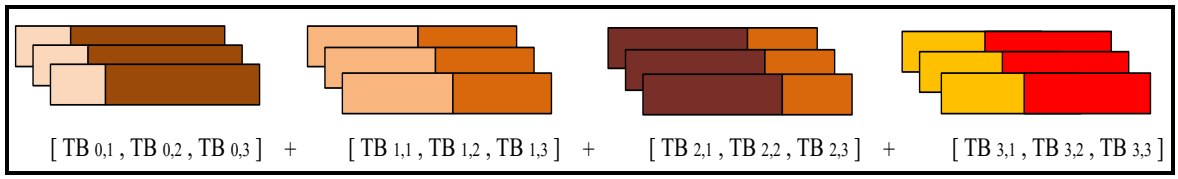


Figure 7.10: The serving eNB side after receiving other eNBs' 3rd retransmitted TB

An important point that required to be noticed is the overhead of forwarding the received copies of the TBs from the non-serving eNBs of the coordination set to the serving eNB of the coordination set. Since, these eNBs communicate with each other using the X2 interfaces, which have very high bandwidth and very small delay, the overhead of forwarding the copies of the received TBs should be tolerable for the system. However, a deep study is required to evaluate the effect of forwarding the copies of the received TBs among eNBs of the coordination sets on the system performance. This study can be considered as a future work into this thesis.

Moreover, the non-serving eNBs send the analog signal of the received TB to the serving eNB. However, they can send their interpretation from the received signal in form of a MAC layer data unit to the serving eNB as well.

7.2 Summary

In this chapter, we proposed a method to improve the traditional retransmission procedure of the LTE-Advanced protocol stack. The main goal of this method is to prevent unnecessary retransmission of the erroneously received TB by the UE. To do so, in case of an unsuccessful transmission, the proposed method allows the serving eNB of the coordination set of a UE, to communicate with other eNBs of the coordination set to request a copy of a received TB at their side. After receiving those copies, the serving eNB perform soft combing on them. This process increases the probability of the data recovery without taking the UE time.

8 SSU Modeling and Simulation

In the previous sections, we have introduced two advanced algorithms for the user data upload in the distributed CoMP architecture. In this chapter, we focus on the modeling and simulation (M&S) of one of the proposed methods (SSU) to test its correctness and to evaluate its performance using the DEVS tools discussed in Chapter 2. In addition, we study the performance of the SSU algorithm in urban and rural area settings. In order to evaluate the performance of SSU, another the non-cooperative algorithm was simulated. In this method, each UE only interacts with its serving eNB instead of all the eNBs in its coordination set. A UE starts the upload with a request message, similar to the one employed in SSU. Once the eNB acknowledges the request, the UE starts uploading the data file in variable sized packets, which depend on the available bandwidth and data rate for the link. The upload processes ends when the UE sends the eNB a Done message following the last data packet.

8.1 Modeling of SSU in DEVS

A DEVS model for mobile networks and SSU we built consists of the various coupled models and the atomic models, following the structure shown in Figure 8.1. As seen in this figure, the top-level model is called *Area*. This coupled model includes one atomic model, *LogManager*, and three other coupled models (*Atmosphere*, *UEManager*, and *BSManager*) and the interconnections among these models.

The *LogManager* is responsible for gathering statistics during the simulations. The *Atmosphere* coupled model in Figure 8.1 consists of two atomic models and the required interconnections. It models the communication between each pair of BSs and UEs (BS is an implementation of eNB). This model is used to receive all the sent messages from each UE and BS, and broadcasts them to all the other models. The BSs and UEs can then recognize their messages based on the destination address field of the received message. Both the *BSManager* and *UEManager* (as the coupled models) almost have the same

structure. Each of them has a Registry unit (as a DEVS atomic model), which is responsible for some management and control actions for BSs and UEs, respectively.

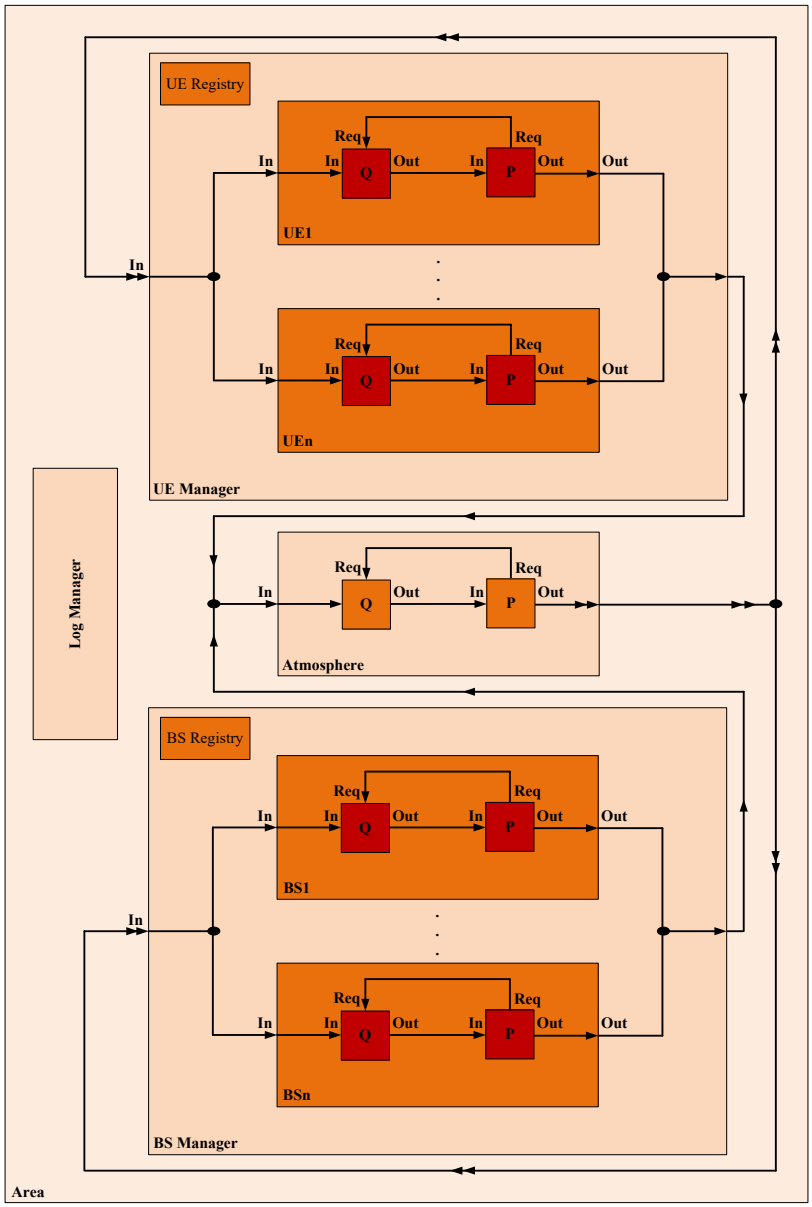


Figure 8.1: Simplified model hierarchy for a mobile network

(Q: Queue, P: Processor)

In addition, the *BSManager* and *UEManager* models can include any number of BS and UEs, depending on the size of the *Area*. The number of UEs in *UEManager* is usually between four or five times the number of BSs. Both UE and BS models are coupled

models, and they are composed of two atomic models: *Queue* and *Processor*. The arrival of a message at a *Queue* is processed based on the message's delay time. Therefore, among all the messages in the queue, the one with the least delay time leaves the queue first. The *Processor* of the UEs and the BSs operate based on the definition of SSU. In the SSU DEVS model, a *UEProcessor* only handles the upload of one file at a time, and therefore, its state transitions directly correspond to the different steps of the proposed algorithm. Figure 8.2 shows a DEVS state diagram for this atomic model. This figure explains the overall operation of *UEProcessor* atomic model once the UE intend to upload a data file. According to this figure, the model is initially *idle*, and it starts the upload process by changing its state to the *Create and Upload* state. To complete the upload process using the SSU algorithm, *UEProcessor* cycles through the states in this figure, which, as said, follow the SSU algorithm defined earlier.

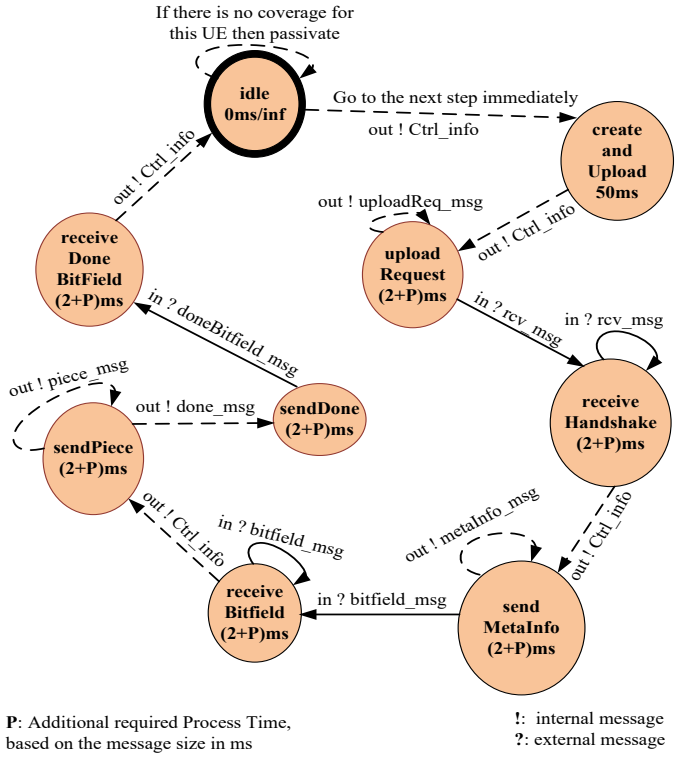


Figure 8.2: UEProcessor DEVS state diagram

On the other hand, a *BSProcessor* handles incoming and outgoing messages from/to neighboring BSs and multiple UEs. Its state transitions cycles through four states,

namely, *Idle*, *Receive*, *Process*, and *Send*, once for each external message received. Figure 8.3 shows a DEVS state diagram for *BSProcessor*. For example, when a BS in *Idle* state and receives an upload request, it replies with a *Handshake* message, and it changes to the *Idle* state again. Thus, the behavior of the model depends on the state of the *UELink* between the BS and the corresponding node at which the received message originated. *UELink* is not an Atomic DEVS model. Rather, it is just a data structure, which keeps track of events between each pair of UE and BS. The *UELink* states are *idle*, *receiveUploadRequest*, *sendHandshake*, *receiveMetaInfo*, *sendMetaInfo*, *receivePiece*, *sendPiece*, *receiveDone*, and *sendBitField*.

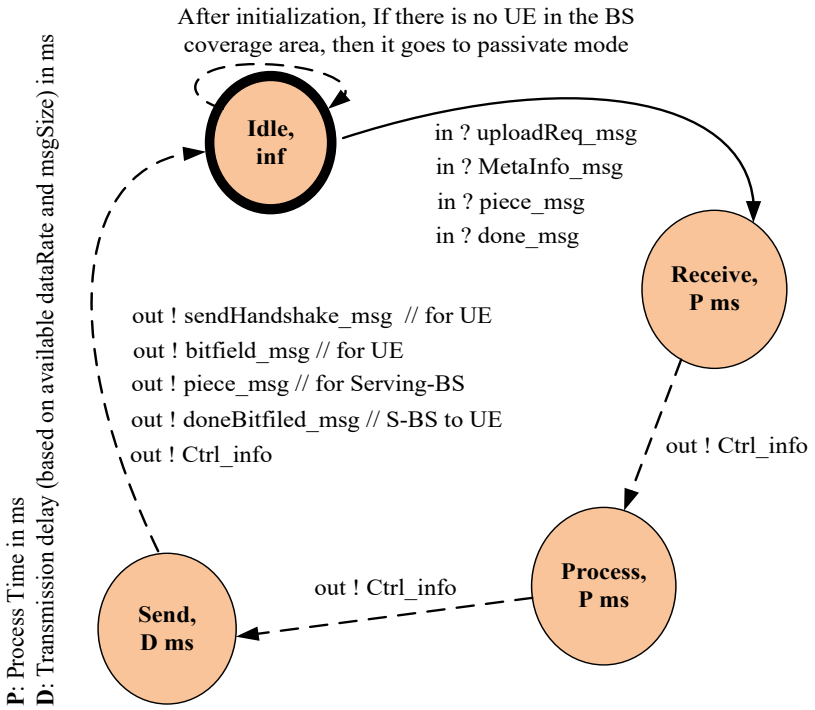


Figure 8.3: BSProcessor DEVS state diagram

Aside from the atomic model components described earlier, a few other passive classes have been added to complete the model. A simplified UML class diagram presenting some of these classes is shown in Figure 8.4, Figure 8.5 and Figure 8.6. Figure 8.4 illustrates some of the required passive classes and the relation among these classes. Figure 8.5 and Figure 8.6 represent the detail of some of these classes including their functions and parameters.

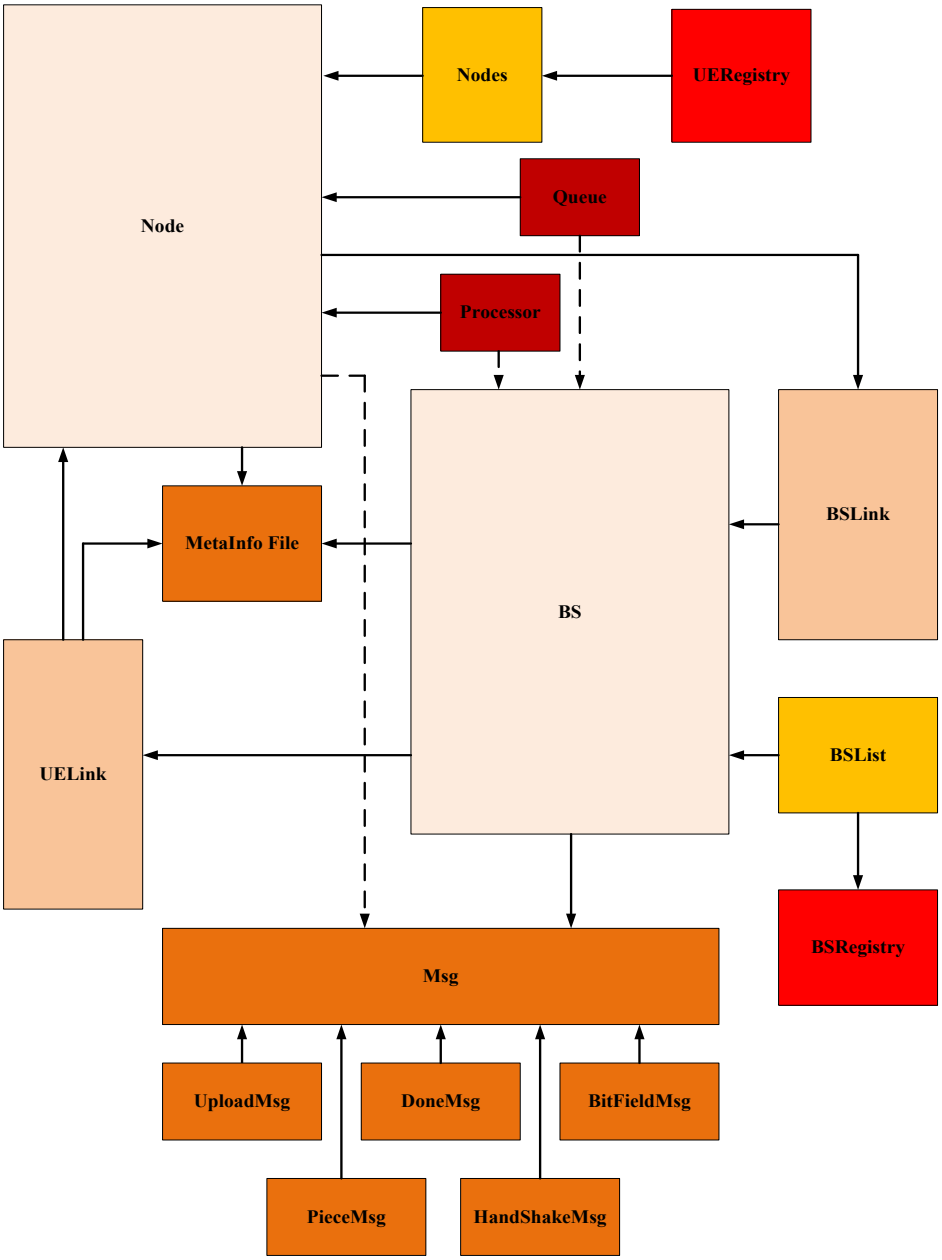


Figure 8.4: Simplified class diagram of the model (1)

The *UELink* class defines a list held by every BS that contains the downlink parameters of the communication link to each of the UEs within range. These parameters include the separation distance, path loss, and the received power. Similarly, the *BSLink* class is a list held by every *Node* object, and it contains parameters similar to those in the *UELink* class for the uplink connection. The two respective classes have methods to calculate link

parameters such as propagation model, path loss, and the received power in rural area settings. The *Nodes* and *BSLink* classes hold pointers to the head of linked lists of UEs and BSs in the area respectively.

```
Node
……..
- bitField: short int*
- bitFieldSize: int
+ BSLinkHead: BSLink*
- dataFile: FILE*
- id:Long
- field: int
- maxPWR: int
- metaInfoFile: MetaInfoFile*
- minPWR: int
+ next: Node*
- posx: unsigned long
- posy: unsigned long
- destx: unsigned long
- desty: unsigned long
- speed: State
- noisePWR: int
- noiseFig: int
- state: State
- status: bool
- TX_G: int
- TX_PWR: int
……..
+ addBSLink(BSLink *): void
+ createAndUploadFile(): void
+ deActivate(): void
+ deActivateAllBSLinks(): void
+ findBSLink(BS*): BSLink*
+ getID): long
+ getTX_G(): int
+ getTX_PWR(): int
+ reActivate(): void
+ receiveBitFieldMsg(BS*, BitFieldMsg*): void
+ receiveHandshakeMsg(BS*, HandshakeMsg*): void
+ removeAllinactiveBSLinks(): void
- sendUploadRequest(UploadRequest*, BS*): void
- sendMetaInfoFile(MetaInfoFile*, BS*): void
- sendPieceMsg(PieceMsg*, BS*): void
- sendDoneMsg(DoneMsg*, BS*): void
+ startSendingPieces(): void
+ setDestination(unsigned long, unsigned long): void
+ setPosition(unsigned long, unsigned long): void
+ setSpeed(double): void
+ setStatus(bool): void
+ startSendingPieces(): void
+ Node(long, unsigned long, unsigned long, unsigned long,
        unsigned long, double, bool, int)
+ Node(Node&)
+ ~Node()
```

```
BS
……..……..
+ connectedBSlist: list<BSSLink>
- Dhb: int
- Hb: int
- f: long
- id:Long
- max PWR: int
- maxPWRDL: int
+ next: BS*
- noiseFig: int
- noisePower: double
- posx: unsigned long
- posy: unsigned long
- UELinkHead: UELink*
- status: bool
- TX_G: int
- TX_PWR: int
- type: Type
……..…….
+ addUELink(UELink *): void
+ deActivateAllUELinks(): void
+ findUELink(Node*): UELink*
+ getID(): long
+ getTX_G(): int
+ getTX_PWR(): int
+ receiveDoneMsg(Node*, DoneMsg*): void
+ receiveMetaInfoFile(Node*, MetaInfoFile*): void
+ receivePieceMsg(Node*, PieceMsg*): void
+ receivePieceMsg(Node*, BS*, PieceMsg*): void
+ receiveUploadRequest(Node*, UploadMsg*): void
+ removeAllinactiveUELinks(): void
+ sendBitFieldMsg(BitFieldMsg*, Node*): void
+sendHandshakeMsg(HandshakeMsg*, Node*): void
+ setPosition(int, int): void
+ setStatus(bool): void
+ BS(long, unsigned long, unsigned long, bool,
        Type, int, int, long, int)
+ ~BS()
```

```
MetaInfoFile
……...……..
- length: int
- name: string
- pieceLenght: int
- piece: unsigned char*
……..……...……..
+ getLength(): int
+ getName(): string
+ getPieceLength(): int
+ MetaInfoFile(int, string, int, unsigned char*)
+ ~MetaInfoFile()
```

Figure 8.5: Simplified class diagram of the model (2)

The UE and the BS parameters such as transmission power, gain, UE speed, BS antenna height and operating frequency can be initialized to specific values in the Model file to create a simulation scenario. These parameters can also be set automatically by defining the area type in the model file to either rural or urban simulation scenarios. This allows us to change the simulation scenario rapidly for different simulation executions. Other simulation properties such as BS power, noise figure and noise power are automatically set by choosing one of the communication standards defined in [34].

Figure 8.6: Simplified class diagram of the model (3)

UERegistry, an atomic component in the *UEManager* coupled model, is triggered in certain time intervals to update the state of the wireless network. It makes use of the list

of UEs in Nodes to update their current position based on the previous position, and the predefined destination coordinates and speed, and the elapsed time since the last update. Moreover, *UERegistry* periodically updates the status of the communication channels between UEs and BSs. The status includes the validity of the links depending on the distance between the corresponding UE and BS, as well as the uplink and downlink channel parameters discussed earlier.

A class hierarchy for messages has been implemented to encapsulate the contents of the messages used by SSU and allow the model components to communicate with fewer messages. *Msg* objects include IDs of the source and destination components as well as the size of the message object and its type. Subclasses of the *Msg* super class define fields specific to the message, as defined by the algorithm.

Figure 8.7 shows a simplified segment of a coupled model file representing the model hierarchy for the SSU system (based on the one presented in Figure 8.1). The Model file is used to define a DEVS coupled model and its hierarchical structure. In the Model file, coupled models list their components and links between them, and atomic models can list some/all their parameters.

As seen in Figure 8.7, the model hierarchy description is started by defining the *Area* coupled model as the top model. All the components of the *Area* and the required interconnection among these sub models are based on Figure 8.1. The *LogManager*, *Atmosphere*, *UEManager* and *BSManager* are defined in as the second level of DEVS model hierarchy. The Model file includes the parameters passed to the DEVS atomic model. After the *LogManager*, we see the *Atmosphere* coupled model. This coupled model includes two atomic components (*AtmosphereQueue* and *AtmosphereProcessor*) and their interconnections. We also define other components models of the top model. These coupled models themselves may include one or more atomic/coupled models, which are defined in the Model file. By following these steps until defining all models, we can implement the hierarchical structure of our model from Figure 8.1 in a Model file (Figure 8.7).

```
[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]
bsCounter : 16       ueCounter : 64
...
[atmosphere]
Components: atmosphereQueue@AtmosphereQueue
Components: atmosphereProcessor@AtmosphereProcessor
in : in         out : out
Link : in                        in@atmosphereQueue
Link : req@atmosphereProcessor   req@atmosphereQueue
Link : out@atmosphereQueue       in@atmosphereProcessor
Link : out@atmosphereProcessor   out
...
[UEmanager]
components : UEregistry@UERegistry UE1 UE2... UE64
...
[UEregistry]
areaConfiguration : rural
...
[UE1]
components: UE1Queue@UEQueue   UE1Processor@ UEProcessor
...
[BSmanager]
components : BSregistry@BSRegistry  BS1 ... BS16
...
[BSRegistry]
...
[BS1]
components : BS1Queue@BSQueue BS1Processor@ BSProcessor
...
[BS1Processor]
BSId : 1
areaConfiguration : 0      #0 = rural, 1 = urban.
...
```

Figure 8.7: Model file of DEVS model in 8.1

8.2 Simulation and Results

In this section, the performance of SSU algorithm is measured in the urban and the rural area setting. As discussed earlier, DEVS is employed as the M&S technique. The hierarchical nature DEVS allowed us to capture precise information from different levels of the model. In order to evaluate the performance of the SSU algorithm, another the non-cooperative algorithm was implemented. In the non-cooperative method, each UE only interacts with its serving eNB instead of all the eNBs in its coordination set. A UE starts the upload with a request message, similar to the one employed in SSU. Once the eNB acknowledges the request, the UE starts uploading the data file in variable sized packets, which depend on the available bandwidth and data rate for the link. The upload processes

ends when the UE sends the eNB a *Done* message following the last data packet. This algorithm is used to compare the SSU algorithm with an algorithm that does not use the cooperation among the eNBs during the UEs upload process.

8.2.1 List of Parameters

The simulations are carried out in both a rural area and an urban area setting. 900 MHz is used as the operating carrier frequency in the rural setting. In the urban area setting, two operating carrier frequencies are considered: 900 MHz and 2000 MHz. 5 MHz is used as the transmission bandwidth for the both area setting. In addition, the noise density is assumed to be fixed at -174 dBm/Hz, and the log-normally distributed shadowing (LogF) is set to a standard deviation of 10dB. The other detailed simulation parameters are listed in Table 8.1 and Table 8.2 [77]. Table 8.1 includes the required simulation parameters for the rural setting. In case of the urban setting, some of the parameters change.

Parameter	Value
Frequency	900 MHz
Transmission bandwidth	5 MHz
Noise Density	-174 dBm/Hz
MCL (R)	80 dB
eNB Antenna Gain	15 dB
eNB Antenna Height above rooftop (Dhb)	15 m
eNB Antenna Height above ground (Hb)	45 m
LogF	10dB
File size	0.5MB-64MB
Maximum eNB power	43 dBm
Maximum power per DL traffic channel	30 dBm
Minimum eNB power per user	15 dBm
eNB Noise figure	5 dB
Maximum UE power	21 dBm
Minimum UE power	-50 dBm
UE Noise figure	9 dB

Table 8.1: Simulation parameters in a rural setting

Parameter	Value	
Frequency	900 MHz	2000 MHz
eNB Antenna Gain	12 dB	15 dB
MCL	70 dB	

Table 8.2: Simulation parameters in an urban setting

8.2.2 Formulas for the Performance Evaluation

The *UERegistry* atomic model is responsible for periodically updating the UEs' locations based on their current locations, their predefined random destinations and speeds. This periodically updates the propagation model (L) for the links between each pair of eNBs and UEs. The updated propagation model is required to calculate received signal power at the receiver side. Then, the available data rate at the link between UEs and eNBs can be calculated. The following formulas show the required steps to calculate link data rate under both rural and urban setting. Let us assume that R is the eNB-UE separation in kilometers, f is the carrier frequency in MHz, Dhb is the eNB antenna height in m, measured from the average rooftop level and Hb is the eNB antenna height above ground (in m). Then the Macro cell propagation model for rural and urban area is given by Equation 8-1 and Equation 8-2 [77].

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

Equation 8-1

$$\begin{aligned} L_{urban} = {} & (40 * \left(1 - (4.10^{-3} * Dhb)\right) * \log_{10} R) - (18 * \log_{10} Dhb) + (21 * \log_{10} f) \\ & + 80dB \end{aligned}$$

Equation 8-2

Considering the log-normally distributed shadowing (LogF) with standard deviation of

10dB, the path loss is given by Equation 8-3, and the received signal power at each UE and eNB is calculated by Equation 8-4 [77].

$$\text{pathloss} = L_{\text{rural/urban}} + \text{LogF}$$

Equation 8-3

$$R_{\text{pwr}} = T_{\text{pwr}} - \text{Max}(\text{pathloss} - T_{\text{Gain}} - R_{\text{Gain}}, \text{MCL})$$

Equation 8-4

where R_{pwr} is the received signal power, T_{pwr} is the transmitted signal power, T_{Gain} is the transmitter antenna gain, R_{Gain} is the receiver antenna gain and MCL is the minimum coupling loss. The link data rate can then be calculated taking into account the Additive White Gaussian Noise (AWGN), using the Equation 8-5, where B is the transmission bandwidth and N_0 is the noise variance.

$$\text{data rate} = B\ \log_2(1 + \frac{R_{\text{pwr}}}{N_0 \times B})$$

Equation 8-5

8.2.3 Simulation Scenarios and Results

In the first set of simulations presented here, we used 17 eNBs to provide radio coverage over a geographical area of 2800 x 3000 m under urban setting. However, our model is flexible to use different area sizes as well as different number of eNBs and UEs. There are 64 active UEs and each of them uploads one file during the simulation. In each simulation, the UEs are located at a predefined distance range from their serving eNBs. The width of this distance range in which UEs are located initially is 50 m. This means that in the first simulation the UEs are randomly distributed within the first 50 m around their serving eNBs, and in the second iteration, they are randomly located between 50 and 100 m from the serving eNB, and so on (Figure 8.8). This way, we are able to study the

effects of upload algorithms on the UEs performance while the UEs' distance from their serving eNBs increases. Each simulation continues until all the UEs complete their upload. We also assumed a UE only moves inside its host cell during the upload process (to make sure no handover is required), and it selects its serving eNB based on the best-received signal among the eNBs. We considered all the active UEs parameters (upload time, distance from serving eNB, etc.) in the calculation of the average numbers. For example, in order to calculate the average number of connected eNBs during the upload process, if there are ten UEs that want to upload a data file, we add the number of the connected eNBs to each of them together and then we divide the final number by ten.

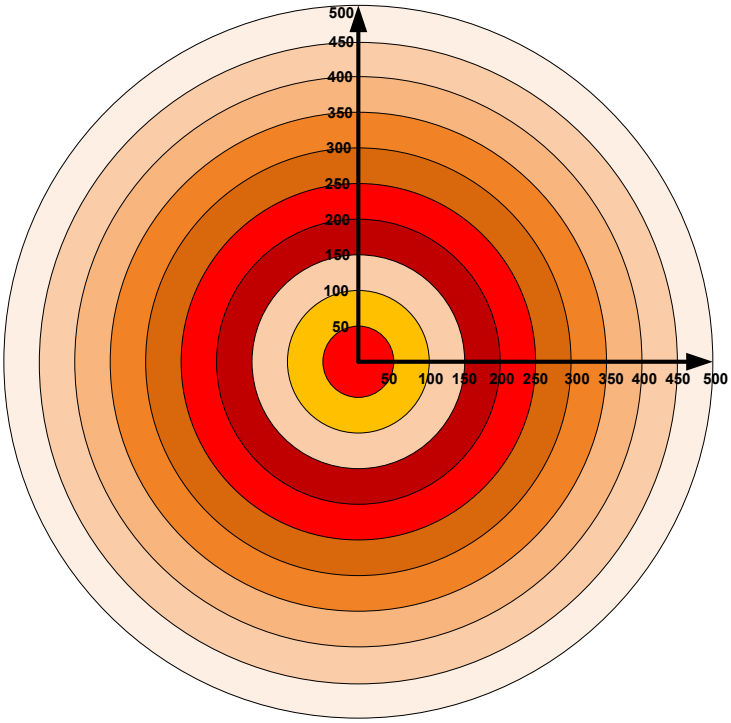


Figure 8.8: UEs are located at a predefined distance range from the cell center

In simulations where the UEs use the SSU algorithm to upload their data, they use their serving eNB to upload their file while they are within a reasonable distance from it. As this distance increases, they may use a set of eNBs to upload their file. In simulations where the UEs use the non-cooperative algorithm, regardless of the UEs' distance, they just communicate with their serving eNB. In Figure 8.9 to Figure 8.14, the horizontal axis

shows the average UEs distance from their serving eNBs. Figure 8.9 and Figure 8.10 show the average number of eNBs that each UE communicates with during the uploading process for urban configuration with 900 MHz and 2000 MHz as carrier frequency.

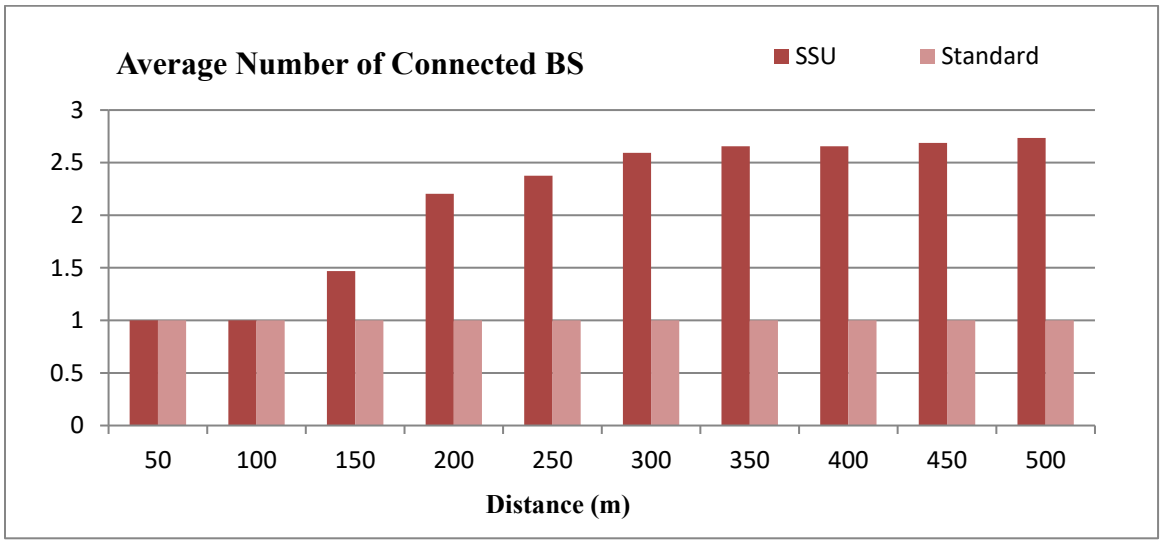


Figure 8.9: Average number of connected eNBs (900 MHz)

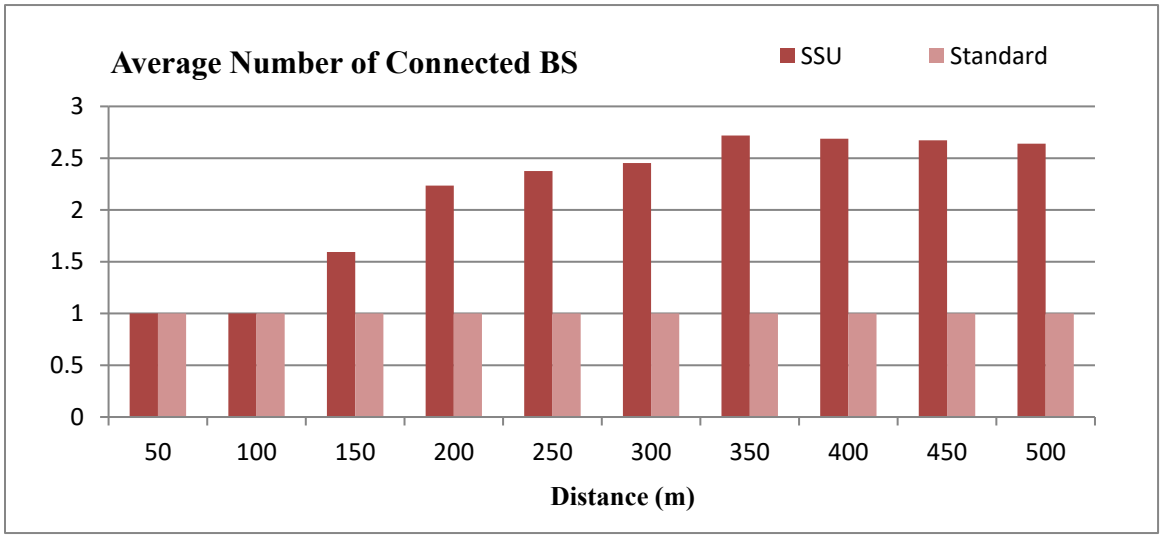


Figure 8.10: Average number of connected eNBs (2000 MHz)

If a UE uses the non-cooperative algorithm for the upload process, it only communicates with its serving eNB during the upload process regardless of its distance from the cell center. In case of SSU, as the UE distance from the serving eNB increases, the UEs can

to communicate with more than one eNB. That is why the average number of the connected eNBs grows up to a certain distance as the distance increases. This growth rate has two types of effects. Using SSU, means that the cell-edge UE potentially communicates with more eNBs and these talk to each other and exchange more control and data messages as well (compared with the non-cooperative algorithm). This means that SSU puts more overhead on the mobile network backhaul (this is a drawback of SSU). The rest of the charts in this section show that there is some overhead, but at the same time, it increases the cell-edge UE performance and reduces the upload time of the data file.

Figure 8.11 and Figure 8.12 show the average upload time as a function of distance (for SSU and the non-cooperative algorithm). The upload process starts with the *Upload Request* message from the UE and it ends when the UE receives the *BitField* message from its serving eNB. Figure 8.11 and Figure 8.12 show that as the UEs' distance from the cell center increases, the non-cooperative algorithm imposes higher delay to its users. This means that SSU provides a better performance for its users. The effects of SSU can be seen more clearly, when the UEs are about 300 to 500 m away from the cell center.

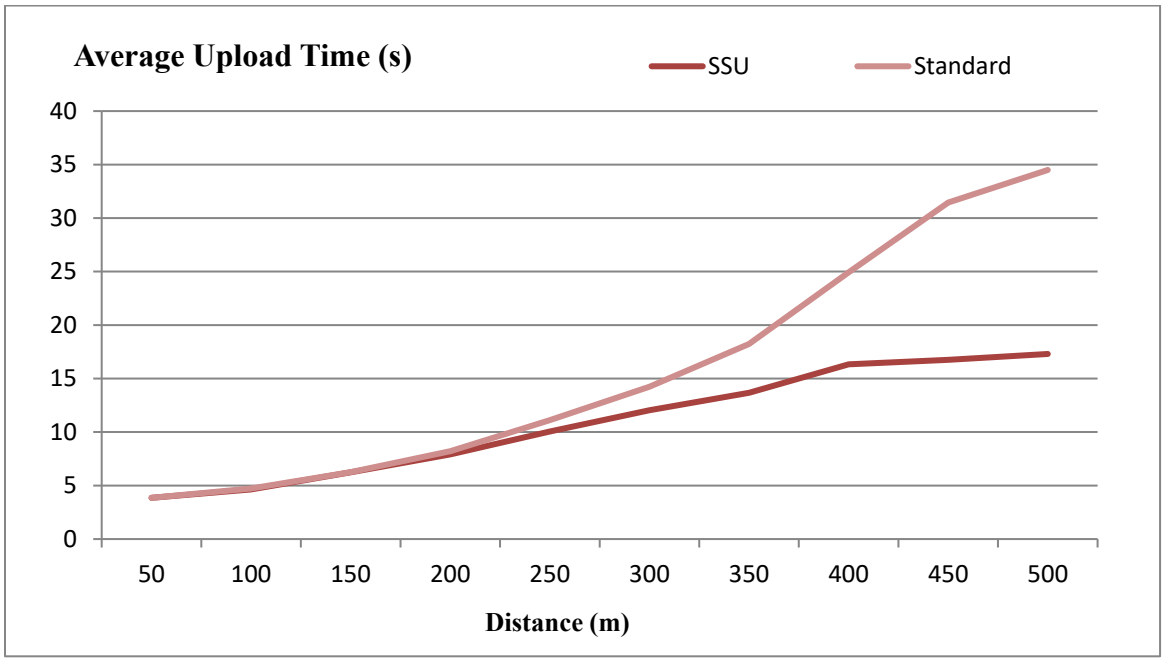


Figure 8.11: File upload time vs. Distance (900 MHz)

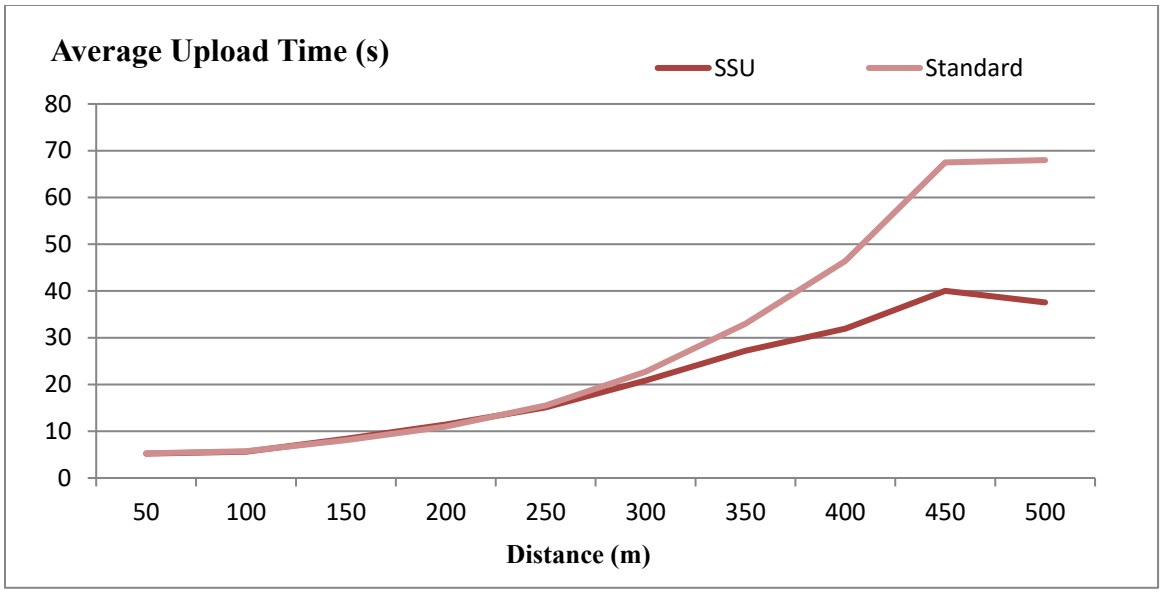


Figure 8.12: File upload time vs. Distance (2000 MHz)

Figure 8.13 and Figure 8.14 show the comparison between SSU and the non-cooperative algorithm with respect to their average data. We can see that the average data rate of SSU for its cell-edge users (located around 500 m from the cell center) is almost double the average data rate that the non-cooperative algorithm provides.

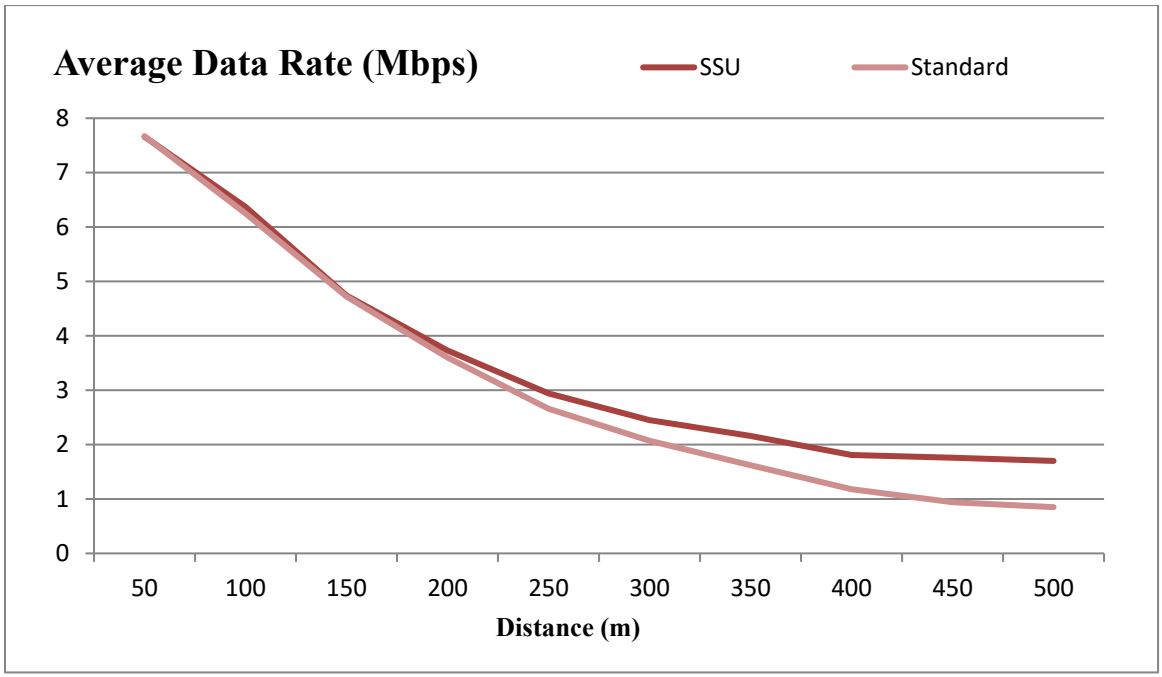


Figure 8.13: Data rate vs. Distance from eNB (900 MHz)

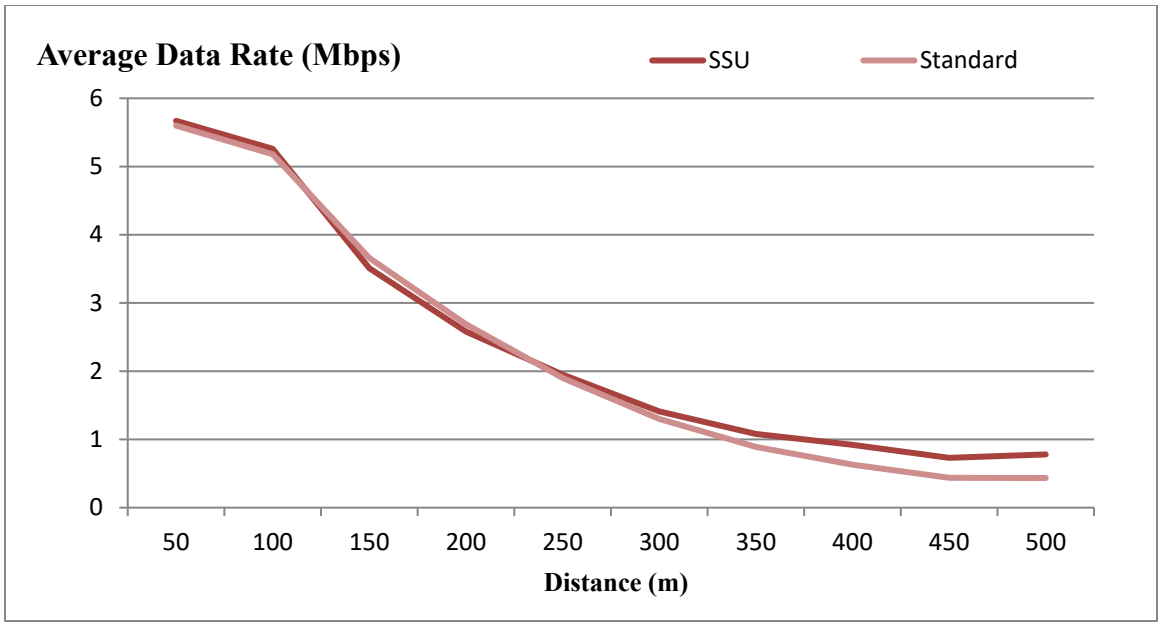


Figure 8.14: Data rate vs. Distance from eNB (2000 MHz)

In the second set of simulations, the designated urban area required 16 eNBs to provide coverage for the entire area. The main goal of this set of simulations is to study the effect of increasing the number of cell-edge UEs on the performance of SSU. We considered a limited scale scenario. We began with 16 cell-edge UEs in the first round, and we increased the number of the UEs on each of the following iterations by 16. In the last round of simulations, we had 160 active cell-edge UEs. In a CoMP scenario, as the number of the cell-edge UEs increases, the number of the required coordination sets increases as well. This means that an eNB may need to deal with more UEs (the UEs belong to this eNB cell and the UEs from neighboring cells) compared to the non-cooperative algorithm (in which each eNB only deals with the UEs of its cell). The increase in the number of coordination sets imposes an overhead on the processing resources (eNBs) and on the backhaul (X2 links). Therefore, adding more coordination sets not always enhance the performance of the system.

In Figure 8.15 to Figure 8.18, the horizontal axis shows the number of the UEs in the coverage area. Figure 8.15 shows that SSU helped the cell-edge UEs upload their data in a shorter period compared to the standard method. In addition, SSU provided better services for its users even when the number of UEs increased.

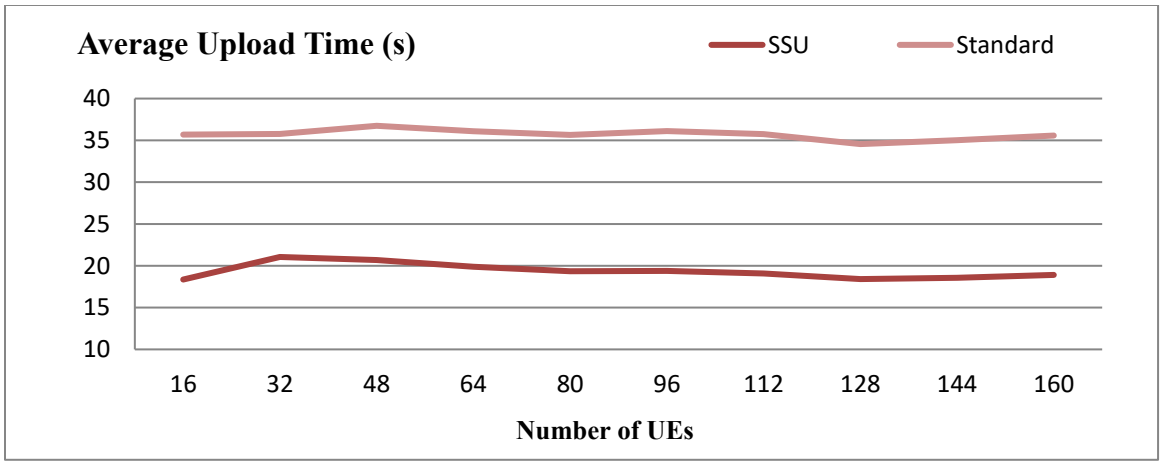


Figure 8.15: Upload time vs. number of UEs (900 MHz)

Similarly, in Figure 8.16, SSU provided a higher data rate for its users and maintained its quality of service while the number of UEs increased. There is some variability in the chart because of the random initial positions of the UEs. Let us assume that at a certain simulation time we add more UEs that are closer to the cell center. This event reduces the average distance of the UEs from the cell center, since we take into account the newly added UEs when we want to calculate the new average distance. On the other hand, we assumed that in our limited scale scenarios, the eNBs have enough resources to support all the UEs. Therefore, when eNBs have enough resources for all the UEs and adding more UEs reduces the average UE-to-eNB distance, then the average data rate increases.

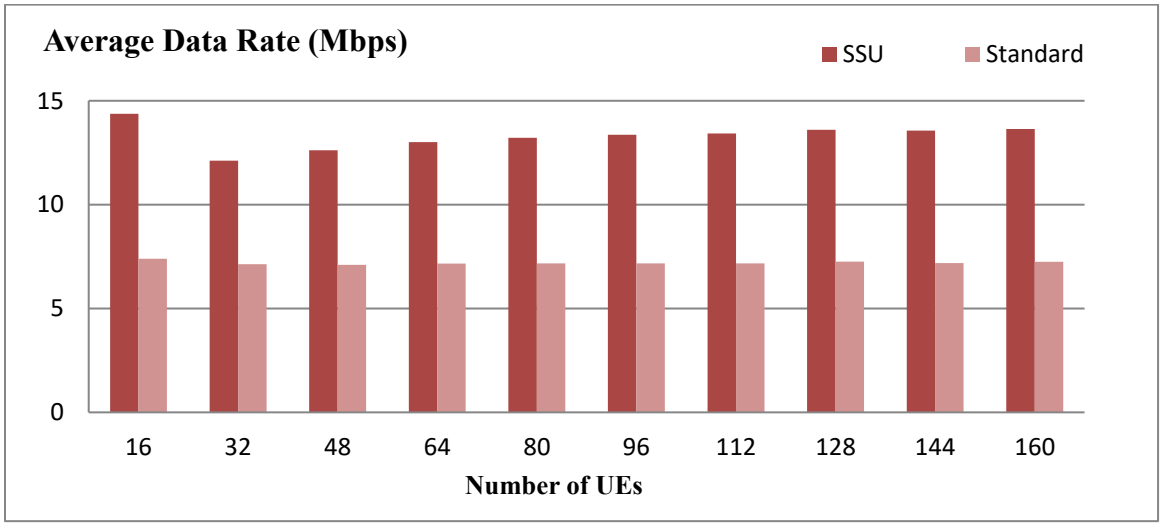


Figure 8.16: Data rate vs. number of UEs (900 MHz)

Figure 8.17 and Figure 8.18 provide similar conclusions regarding the performance of SSU. These figures show that SSU provides lower latency and higher data rate for its users, compared to the non-cooperative method and it could offer an equivalent quality of service even when we increased the number of the UEs in the coverage area. Although these simulations show the effectiveness of SSU in the limited scale scenarios, we need to run more simulations with large number of UEs to evaluate the performance of the SSU as well as its effects on the backhaul in more detail.

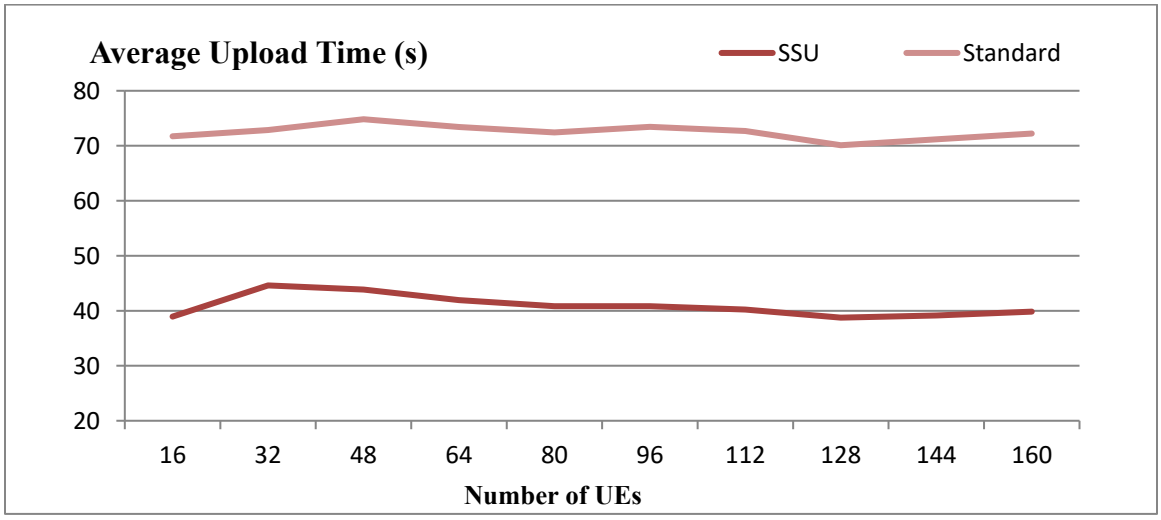


Figure 8.17: Upload time vs. number of UEs (2000 MHz)

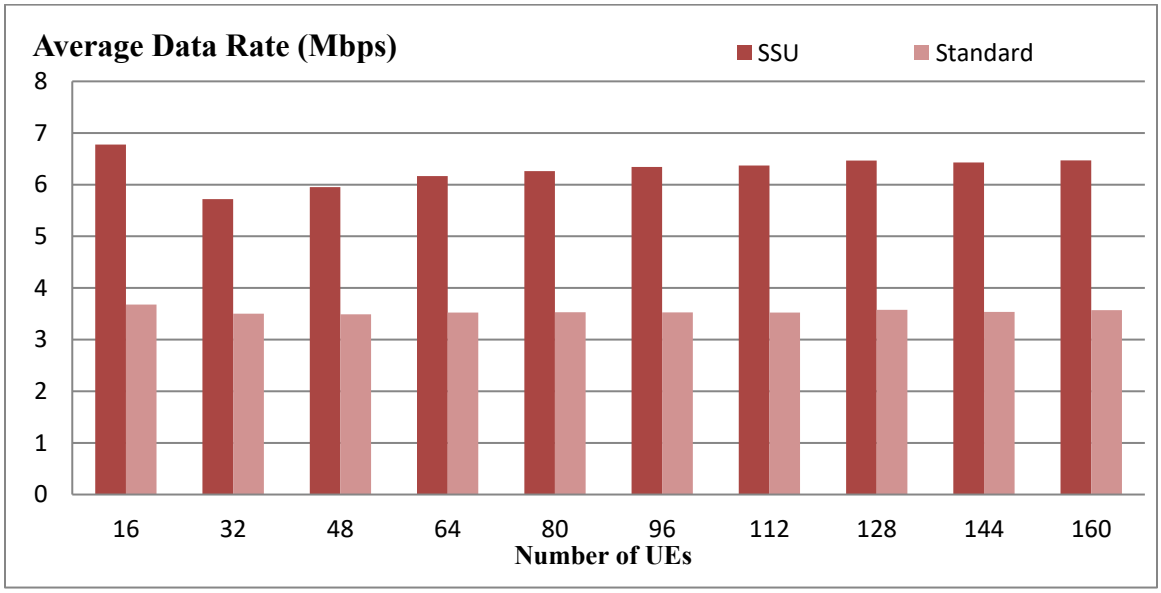


Figure 8.18: Data rate vs. number of UEs (2000 MHz)

Finally, we run a set of simulations in a rural area setting with an operating area of 8 km by 8 km, and eNB-UE distances increasing in increments of 800 m up to 8 kilometers. In other words, the first simulation positioned UEs within the first 800 m around their serving eNB; the second simulation had UEs placed between 800 and 1600 m from the eNB, and so on. The simulations were allowed to run until all the file uploads were complete and the simulation statistics were collected. These files were then analyzed and some of the chosen results are shown in the figures below.

A user can communicate with its serving eNB when traveling in a cellular network that uses the non-cooperative algorithm. This case is true even when the user is in the cell edge areas. On the other hand, SSU provides higher data rate allowing the UEs to communicate with the eNBs in the coordination set while they are close to the cell edge area. Figure 8.19 shows the average number of eNBs that each UE communicates with during the uploading process. As the distance between UEs and the cell center increases, it is more likely that the UEs receive other eNBs signal. Figure 8.20 shows that SSU uses this fact effectively improves UEs performance.

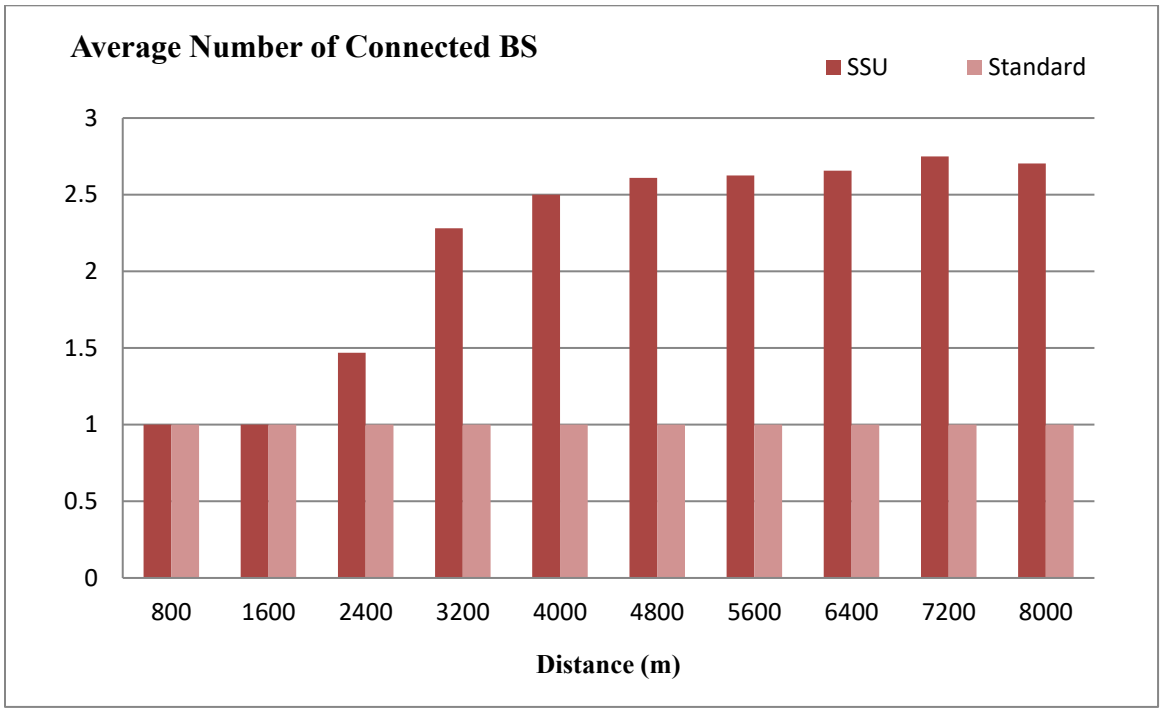


Figure 8.19: Number of connected eNBs vs. Distance from eNB

Figure 8.20 shows the average of file upload time as a function of distance for SSU, as well as the non-cooperative algorithm. The upload time is measured from the moment that the upload process is initiated until the last message is sent. For SSU, it starts with the *Upload Request* message from the UE, and it ends with the *BitField* message from the UE's serving eNB. SSU starts to impact the upload time when the UEs are about 4400 m away from the serving eNB. Closer to the eNB, it adds a small overhead caused by the additional control messages.

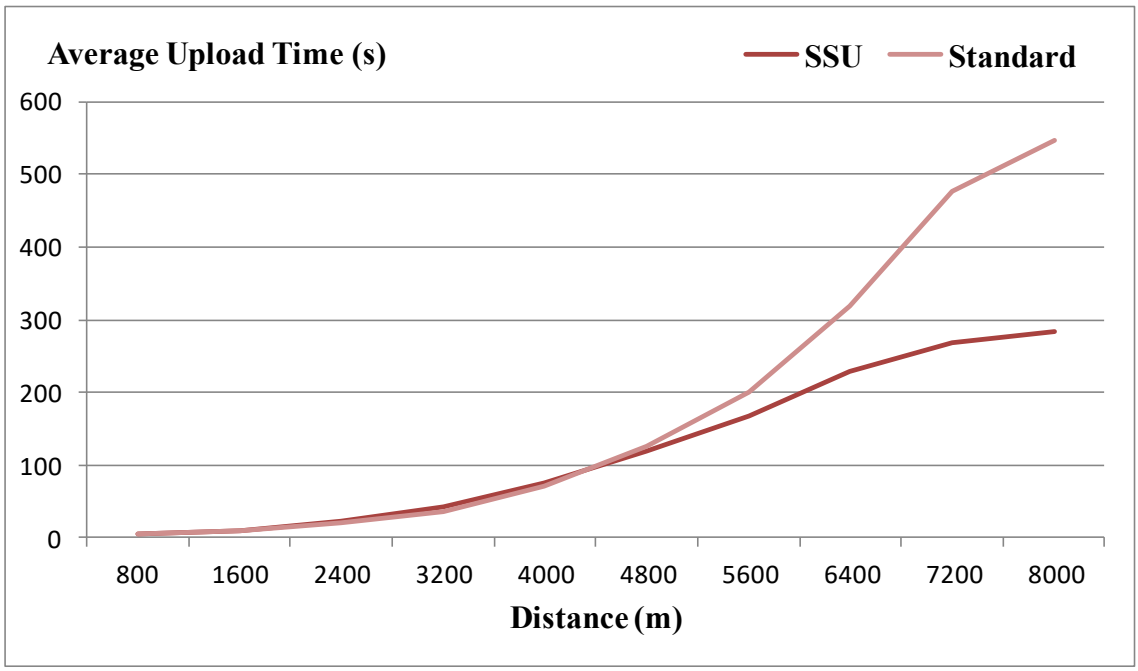


Figure 8.20: Average file upload time vs. Distance from eNB

8.3 Model Verification and Validation

Simulation model Verification and Validation (V&V) refers to the analysis of the system. In this process, the validation phase deals with the accuracy of the conceptual model, and with how well the proposed model addressed a real-world system. The verification phase is used to confirm that the computer model has been correctly implemented in a simulation based on the specifications of the model. The main objective of the process is to insure that the implementation of the model generates the expected output for a given set of inputs.

We developed our DEVS models in collaboration with experts from Ericsson Canada. The experts at the company assisted us with the validation of the mobile network model. We shared the conceptual model along with the assumptions and simplifications with the experts, and discussed the models details. In terms of simulation verification, one of the main characteristic of the DEVS methodology is that the simulation algorithms are based on systems theory and it has been formally proved that the simulation algorithms can execute the DEVS models correctly. This formalization is one of the main advantages of DEVS when compared with other network simulators. The modular and hierarchical nature of the DEVS models also allows testing the model using a bottom-up approach. To do so, we started from the atomic models, and tested different sets of inputs whose results were compared with the outputs generated. Once each component was verified individually for each possible set of inputs, the components were integrated into a larger model, which was verified again in a similar manner. This process repeated until with reached to the top level of our DEVS model.

8.4 Summary

In this chapter, we showed how to model and simulate LTE-A mobile networks using two approaches: Shared Segmented Upload algorithm (SSU) and a non-cooperative method. The simulation results showed that SSU provides services that are more consistent to the users as their distance increases from the cell center. Compared to the non-cooperative method, SSU provided higher data rate for the users and reduced the time to upload its data from a UE to the network. Considering the large amount of data that required to be transmitted over a mobile network, further investigation is required to study its influence on the backhaul.

9 UUC Modeling and Simulation

In chapter 4, we introduced the UUC to improve the cell-edge users upload performance in a distributed CoMP architecture. According to UUC, the UE that owns the data file divides it into a number of pieces at the beginning of the upload. After that, this owner UE starts the upload process by transferring the pieces to the group of eNBs that are coordinated dynamically. At a same time, this owner UE can ask for the help from the neighboring UEs, and let them upload some of the pieces. These UEs can communicate directly with each other using D2D communications.

To investigate the performance of the UUC algorithm, we used DEVS M&S as in the previous chapter. We modeled the UUC algorithm, a CoMP method, and a non-cooperative algorithm. Using DEVS allowed to easily reusing some of the atomic and coupled models of the SSU DEVS model for the M&S of UUC.

9.1 Modeling of UUC in DEVS

One goal in this activity was to design a flexible model for the mobile network and the algorithms. To do so, we separated the functionality of the LTE systems from the behaviour of the algorithms that we wanted to test them. This means that in our proposed model, an entity from the LTE network performs its basic tasks regardless of the algorithm under consideration.

The overall design of the UUC model is similar to the SSU DEVS model in section 8.1. Figure 9.1 shows the top level of the UUC DEVS model. As we can see, the top level is called *Area*, and it includes four coupled models (*MME*, *Atmosphere*, *BSManager*, *UEManager*) and one atomic model (*LogManger*). The *MME* coupled model represents a simplified Mobility Management Entity from the LTE networks. It is connected to all the BSs, and it can send/receive messages to/from them directly. This model composed of two atomic models including *MMEProcessor* and *MMEQueue*. The latter take cares of

the arrival messages until the processor ask for them and the former processes the incoming messages one at a time. The *Atmosphere* coupled model is as same as the one we discussed in section 8.1. *UEManager* and *BSManager* can be composed of various numbers of UEs and BSs respectively. The number of the *BS* coupled models inside the *BSManager* coupled model depends on the number of the BSs that we need to provide the radio coverage over the desired geographical area. The number of the *UE* coupled models is dynamic too, and it depends on the simulation scenario.

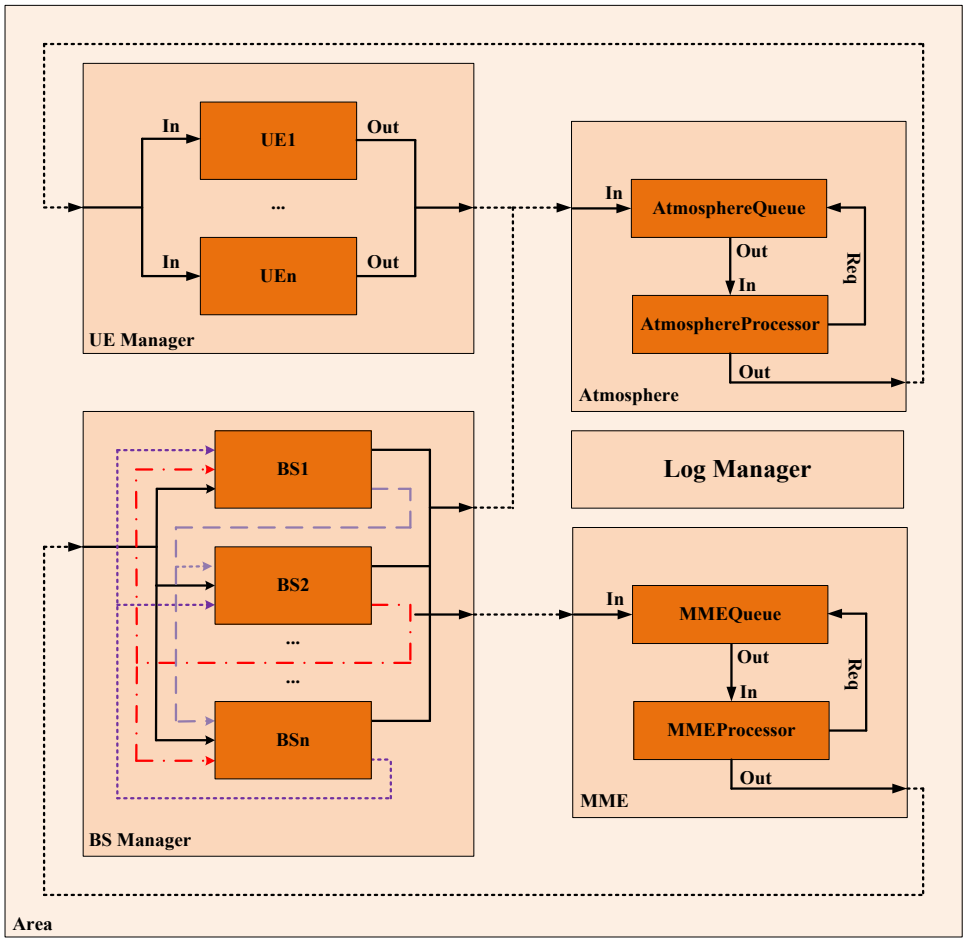


Figure 9.1: Simplified DEVS model hierarchy for a mobile network model

Figure 9.2 and Figure 9.3 illustrate the structure of the *UE* and *BS* coupled models respectively. As seen in Figure 9.2, the UE coupled model contains eleven atomic models and two coupled models. The coupled models are *UEFilter* and *UETransmitter*. *UETransmitter* composed of two atomic models: *UETransmitterProcessor* and

UETransmitterQueue. The latter queue the outgoing messages and the former send them at one-millisecond intervals according to the scheduled resources available to the UE. This means that the *UETransmitter* coupled model synchronizes the outgoing messages sent by the UE with LTE's Transmission Time Interval (TTI). LTE TTI refers to the smallest period used for scheduling resources and transmitting data blocks in a LTE system, and it is equal to one millisecond [35]. The *UEFilter* coupled model has a same structure like the *UETransmitter* coupled model. The *UEFilterQueue* atomic model receives the arrival messages from the *Atmosphere*, and the *UEFilterProcessor* performs required tasks on these messages and passes the messages that belong to this destination to the UE physical layer queue. The arrival messages are broadcast in nature, and at each receiver side (UE or BS), we need to filter them based on their intended destination. As we said, the UE coupled model includes several atomic models, which can be grouped into the three categories. The first category is about the atomic models that represent the simplified behaviour of the different layers of the LTE-Advanced protocol stack. This group includes *UEPhysicalLayer*, *UEMACLayer*, *UERLCLayer* and *UEPDCPLayer* atomic models. We also added a *UEAPPLayer* atomic model into this category to initiate the file upload process at the beginning of the simulations. In the second category, we have an instance of a *UEQueue* per each atomic model in the first category. The primary responsibility of these atomic models is to queue the arrival messages to the each layer. Finally, the last category includes a *UETimer* atomic model that is used to generate a signal to synchronize the processing of arrival messages with LTE's TTI.

As seen in Figure 9.3, the structure of the *BS* coupled model is almost as same as the *UE* coupled model structure. However, *BS* coupled model has more input and output ports as well as more interconnection. A *BS* coupled model can send/receive messages to/from the *MME* coupled model via the correspondent ports (*toMME* and *fromMME*). In addition, there is an input port per each neighbour BS (*fromNeighbourX*), and there is an output port to each neighbour BS (*toNeighbourX*). The number of the neighbours of each of the BSs depends on the position of that BS in the area. A BS may have up to seven neighbours.

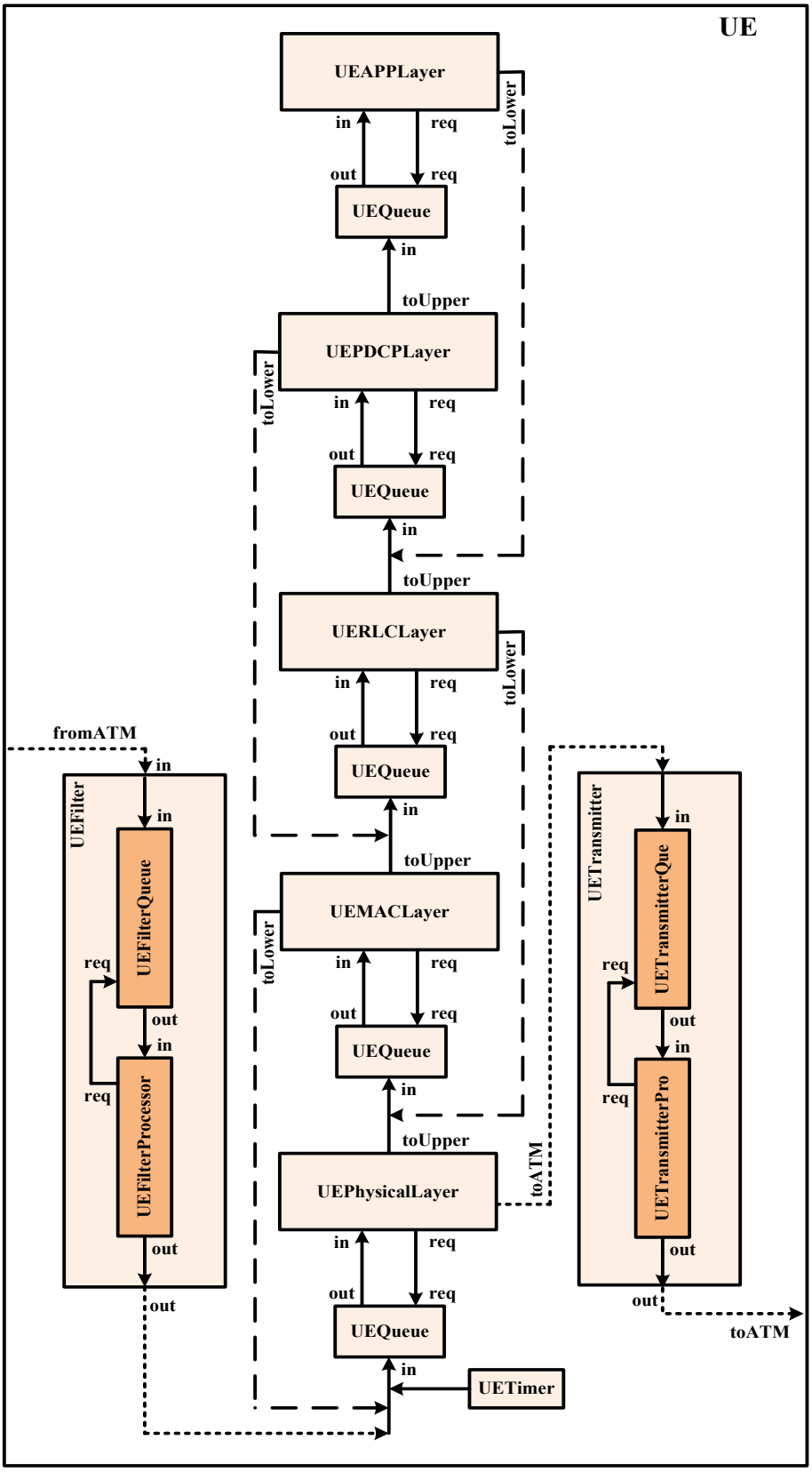


Figure 9.2: Structure of the UE coupled model

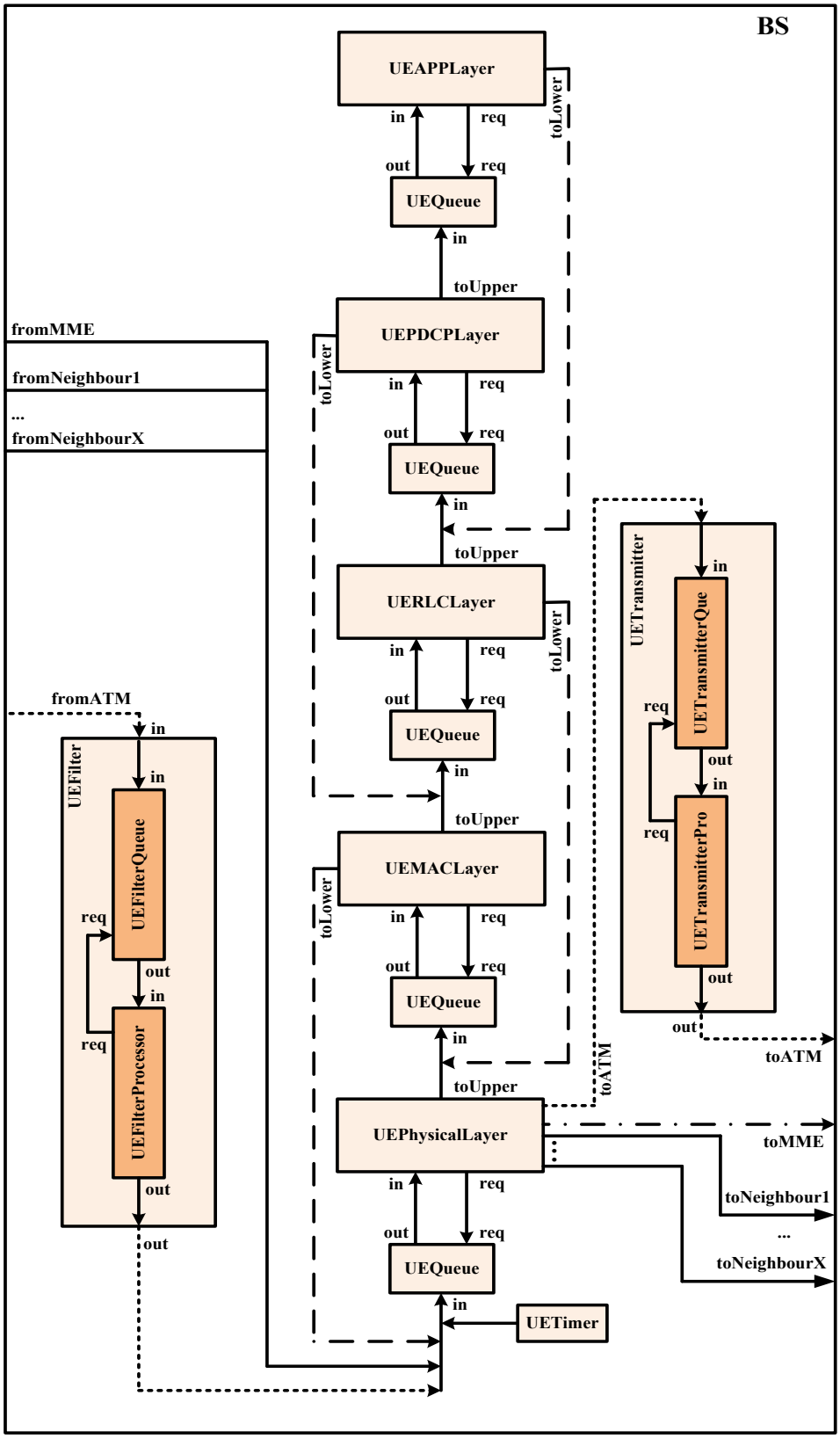


Figure 9.3: Structure of the BS coupled model

In the UUC DEVS model, some of the atomic models correspond to the different layers of the LTE-Advanced protocol stack. Each of these atomic models receives the messages in order, and it processes them, according to the definition of the algorithm. Figure 9.4 shows the DEVS state diagram of these atomic models.

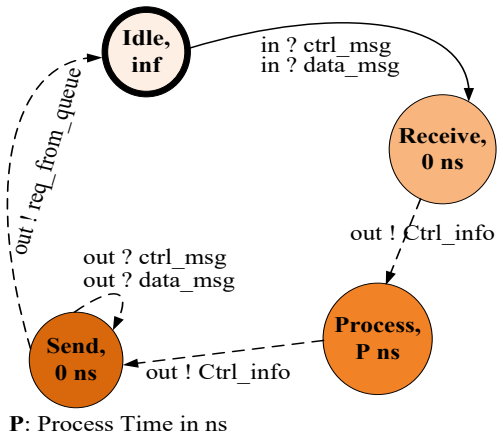


Figure 9.4: DEVS state diagram of a sample atomic model

Idle is the initial state of these atomic models. Upon receiving an external message, they switch to the *Receive* state, where they retrieve the correspondent session to the received message. After that, they switch to the *Process* state to process the message based on the algorithm definition. In this state, they also may generate zero, one, or more than one message as the outgoing messages. As the next step, these atomic models switch to the *Send* state to transmit the outgoing messages one at a time. Finally, these atomic models send a request message to their correspondent queue model to ask for the next arrival message, and they switch back to the *Idle* state.

The atomic models are 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 9.5 shows the Unified Modelling Language (UML) class diagram for the model's atomic classes. Finally, same as the SSU DEVS model, aside from the atomic model components described earlier, a few other passive classes have been added to complete the UUC DEVS model.

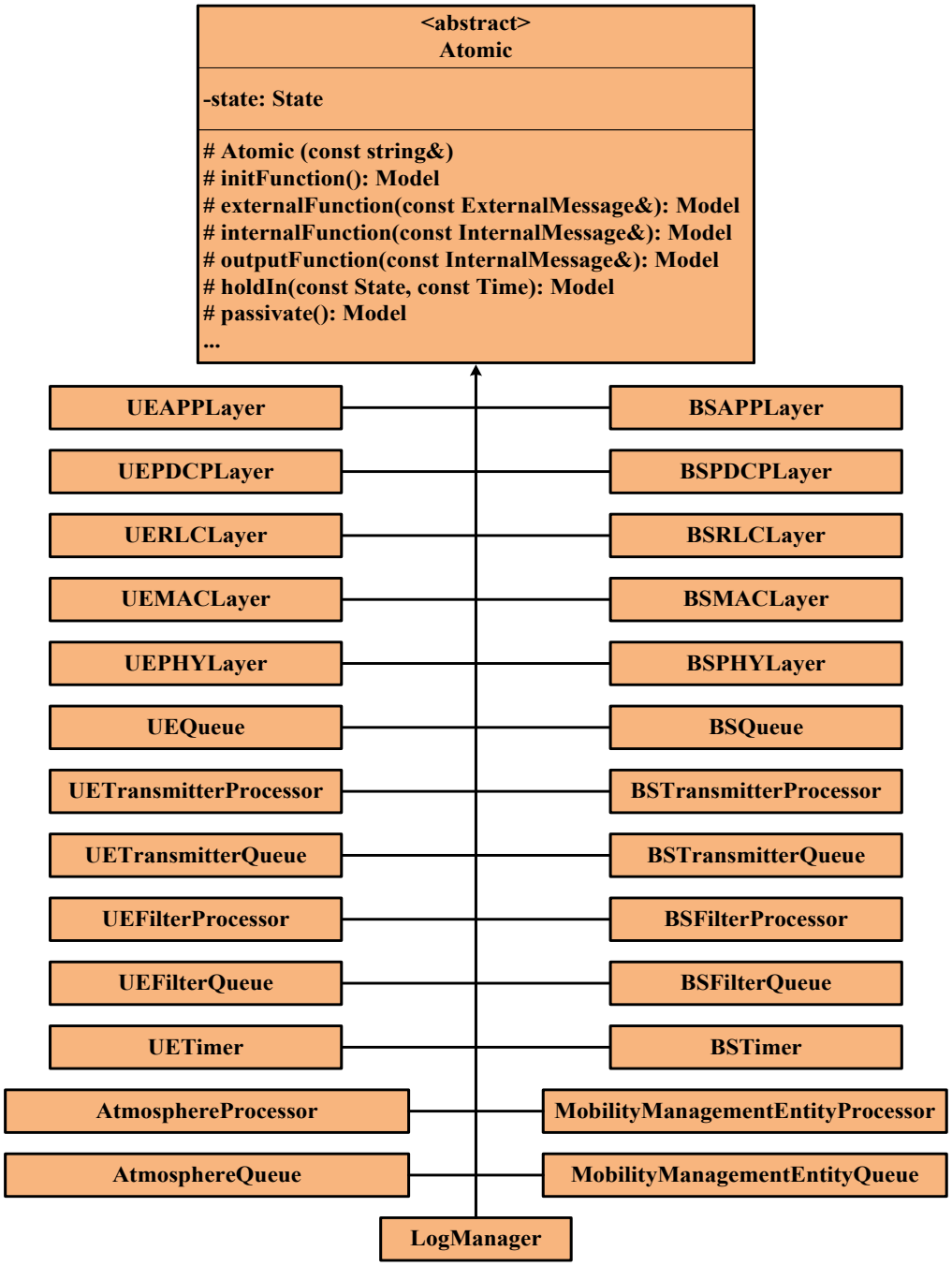


Figure 9.5: Simplified UML class diagram for the atomic model classes.

9.2 Simulation and Results

In this section, we show the performance of UUC in an urban area setting. In order to evaluate the performance of UUC, the non-cooperative algorithm, as well as a model of CoMP was implemented.

In the non-cooperative method, each UE only interacts with its serving eNB instead of all the eNBs in its coordination set. A UE starts the upload with a request message, similar to the one employed in SSU or UUC. Once the eNB acknowledges the request, the UE starts uploading the data file in variable sized packets, which depend on the available bandwidth and data rate for the link. The UE sends the eNB a *Done* message following the last data packet and the upload process ends when the UE receives the *UploadFinished* message from its serving eNB.

The second method is based on CoMP using *Joint Transmission*. In this method, each eNB of the UE coordination set receives its own copy of the transmitted signal. In other words, multiple eNBs, which are geographically distributed in different cells, receive their own copy of the UE signal. This form of joint reception is usually referred to as *Receive Diversity* [78]. Then, the non-serving eNBs forward the received copies to the serving eNB over the backhaul (X2 links). The serving eNB selects the signal from the diversity branch (one of the receiver eNBs) with the highest instantaneous SNR. The latter approach, selecting the signal with the maximum SNR, is referred to as Conventional Selection Combiner (CSC) [79]. As it is clear, in the second method, a UE only communicates with the eNBs in its coordination set, and it does not communicate with neighboring UEs.

When the UE uses Joint Transmission CoMP in the uplink process, the UE distance from the serving eNB has an important effect on the CoMP performance. When a UE is sufficiently close to the cell center (where its serving eNB located), it has a good communication channel with its serving eNB, while the received signal by the non-serving eNBs is likely to be weak (due to the distance between the UE and the non-serving eNBs). Therefore, the serving eNB cooperation with these non-serving eNBs not only reduces the performance gain, but also increases the overhead. To avoid this problem and to prevent 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. This means that the 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 [80][81]. Finally, in the implemented CoMP technique, the UE begins the upload by sending an *UploadRequest* message to the eNBs in range, and it ends the upload when it receives an *UploadFinished* message from its serving eNB.

9.2.1 List of Parameters

All D2D communications operate on a carrier frequency of 2000 MHz [82]. The shadowing for D2D transmissions was calculated using a standard deviation of 7 dB, and the UE antenna height was fixed to 1.5 m. The simulations were carried out in an urban area setting using 900 MHz as the operating carrier frequency. In addition, the log-normally distributed shadowing (LogF) is set to a standard deviation of 8 dB. The other detailed simulation parameters are same to the ones presented in section 8.2.1.

9.2.2 Formulas for Performance Evaluation

We used the propagation model from section 8.2.2 to simulate UUC. The propagation model considered for D2D communication assumes outdoor-to-outdoor communication, which means that both the sender and receiver UE are located outdoors. According to the channel models discussed in [82], the path loss for a direct Line-of-Sight (LoS) and non-Line-of-Sight (NLoS) transmission between two UEs can be calculated using Equation 9-1 and Equation 9-2 respectively [83]. LoS refers to a transmission, where there is no obstacle between the sender and receiver UEs and they are in view of each other.

$$L_{D2D,\ LoS} = (22.7\times \log_{10} d) + 27 - (20\times \log_{10} f)$$

Equation 9-1

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

Equation 9-2

Where f (in MHz) is the operating carrier frequency for D2D transmissions, d (in m) is the UE-to-UE separation distance, and h (in m) is the UE antenna height from the ground

in m. 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 previous two channel models (Equation 9-1 or Equation 9-2) to calculate the overall channel path loss ($\mathrm{pathloss_{B1}}$). The probability is calculated using Equation 9-3 [84]. After that, the channel path loss is calculated according to Equation 9-4 [82], where $\mathrm{pathloss_{fs}}$ is the free space path loss (the minimum path loss a channel may experience), and it is calculated based on Equation 9-5 [84].

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

Equation 9-3

$$\mathrm{pathloss} = \max\ (\mathrm{pathloss_{fs}}, \mathrm{pathloss_{B1}}\ )$$

Equation 9-4

$$\mathrm{pathloss_{fs}} = \left(20 \times \left(\log_{10} \mathrm{d} + 46.4 + \log_{10}\left(\frac{\mathrm{f}}{5.0}\right)\right)\right)$$

Equation 9-5

A Log-normally distributed random variable, $D2DLogF$, was used for the shadowing implementation. In addition, a Rayleigh distributed random variable ($rayleighFading$) was used to model small-scale fading. The resulting propagation losses are combined using Equation 9-6. The received power is calculated based Equation 9-7. Finally, data rate of the transmission link is calculated using Equation 9-8. The latter takes into account additive white Gaussian noise [77].

$$\mathrm{pathloss\_F} = \mathrm{L} + \mathrm{D2DLogF} + \mathrm{rayleighFading}$$

Equation 9-6

$$\mathrm{Pwr_R} = \mathrm{Pwr_T} - \mathrm{Max}(\mathrm{pathloss\_F} - \mathrm{G_T} - \mathrm{G_R},\ \mathrm{MCL})$$

Equation 9-7

$$\mathrm{Pwr_R} = \mathrm{B} \times \log_2(1 + \frac{\mathrm{Pwr_R}}{\mathrm{N_0} \times \mathrm{B}})$$

Equation 9-8

9.2.3 Simulation Scenarios and Results

We performed a number of simulations based on the UUC DEVS model described in the previous sections. This model represents the UUC behavior described in Chapter 4. In addition, it supports the non-cooperative algorithm, as well as CoMP. Moreover, we extended this model to remove the fixed coordination set constraint from the owner and the helper UE. To do so, this model was improved to support the super set models described in section 5.4.1. (this DEVS model still does not support the proposed retransmission and handover schemas, which is left for future work). We ran a number of simulations to reach to a 95% confidence interval.

In the first set of simulations (scenario 1), we used 9 eNBs to provide radio coverage over a geographical area of 1875 m by 2165 m under urban setting. There are up to 70 active UEs depending on the upload algorithm, and each of the owner UEs uploads one file during the simulation. In each simulation, the owner UEs are located at a predefined distance range from their serving eNBs starting from 100 m, and the BS-UE separation distances are increased in each simulation run in increments of 50 m. In addition, the width of this distance range in which UEs are located initially is 10 m. This means that in the first simulation the UEs are within the 100-110 m distance from their serving eNBs, and in the second iteration, they are located between 150 and 160 m from their serving eNB, and so on. This way, we are able to study the effects of all three upload algorithms (non-cooperative, CoMP and UUC) on the UEs performance while the UEs' distance from their serving eNBs increases. We used a uniform distribution to generate the size of the data file of the owner UEs, which is a number between 30 and 60 Mb. Moreover, a piece size of 1 Mb was selected. In all those simulations, the owner UE to helper UE distance is 10 m. Each simulation continues until all the UEs complete their upload. We also assumed that the UEs do not move during the upload process, and they select their serving eNB based on the best received signal from the eNBs.

We conducted eight different simulations per each UE-to-BS distance range. In all of them, the owner UEs upload parameters like the file size and their distance from their serving eNB are same. In the first and the second simulations, each owner UE uploads the file to the mobile network using the non-cooperative and CoMP in order. In the following figures, we refer to the non-cooperative method as the *Normal* method. In the last six simulations, the owner uses UUC for the upload process. The only difference among these six simulations is the number of the helper UEs per each owner UE. In the first one, each owner UE has only one helper UE, and in the second one, each owner UE has two helper UEs and so on. In the following figures, we refer to these simulations as UUC-H1, to UUC-H6.

Figure 9.6 shows the average number of eNBs that participated in the upload of files (per file) versus the average distance of UEs from their serving eNBs. With the non-cooperative algorithm in use, a UE only communicates with its serving eNB resulting in an average number of connected eNB of one. When the UE is close to the center of the cell, CoMP behaves in a similar manner to the conventional algorithm in terms of the number of eNBs the UE communicates with. However, as the UE moves closer to the cell's edge, it is more likely that the UE receives signals of multiple eNBs, and it uses CSC method. This explains why, as seen in the rest of the figures, CoMP provides better performance for UEs close to the cell's edge. In term of UUC, the average number of the eNBs that participate in a file upload depends on the number of the helper UEs as well as the UE-to-eNB distance. Each helper UE communicate with a set of eNBs, while it uploads the pieces, and the number of the eNBs in this set was considered when we calculated the average number of the eNBs in Figure 9.6. It worth to say that we can refer to this figure as it shows the average number of the transmission channels that the UEs use to upload a specific data file. Since, UUC can work on top of CoMP, the average number of the eNBs increases as the UE-to-eNB distance increases. According to Figure 9.6, a UE starts adding more eNBs to its coordination set around 350 m from the cell center in urban areas. It worth to say that the UE could receive signals from other eNB even when it is closer to the cell center, however, the signal quality was not good enough to convince the UE to add that eNB into its coordination set. In addition, it is good to

remind that in the UUC method, an owner UE with a number of helper UEs try to upload the file pieces. For example, in case of UUC-H1, two UEs (an owner UE and a helper UE) in total try to upload the file pieces.

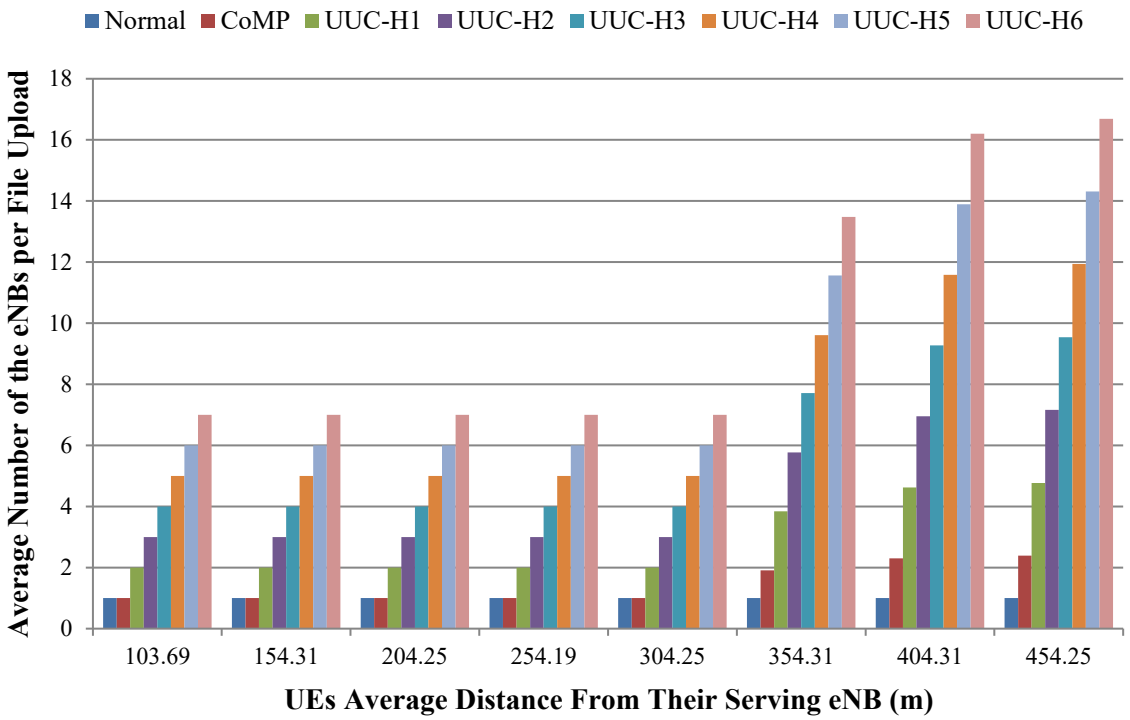


Figure 9.6: Average number of eNBs versus the UE-to- eNB distance

Figure 9.7 shows the average upload time for a data file as a function of distance for UUC, the non-cooperative algorithm and CoMP in an urban area setting. Figure 9.8 is a focused version of the Figure 9.7, and it focuses on the first two seconds of the upload process of all the methods. Figure 9.8 provides clearer picture of the difference among the upload time of the UUC method with various number of helper UEs.

Figure 9.7 shows that the non-cooperative algorithm and CoMP behaved in a similar manner when the UE was within 305 m from the cell center. In both methods, as the UEs distance from their serving eNBs increased, the file upload time increased as well. However, when the UEs got closer to the cell borders, CoMP allowed UEs to upload their data file in a less time compared to the non-cooperative upload time. For example, at around 350 m away from the cell center, the upload time for CoMP and the non-

cooperative algorithm was about 6.19 and 6.52 seconds respectively. These numbers reached to 6.42 and 7.32 in order, when the UEs were about 450 m away from the cell center. This means, compared to the non-cooperative algorithm, CoMP could enhance the cell-edge users' performance by 13%.

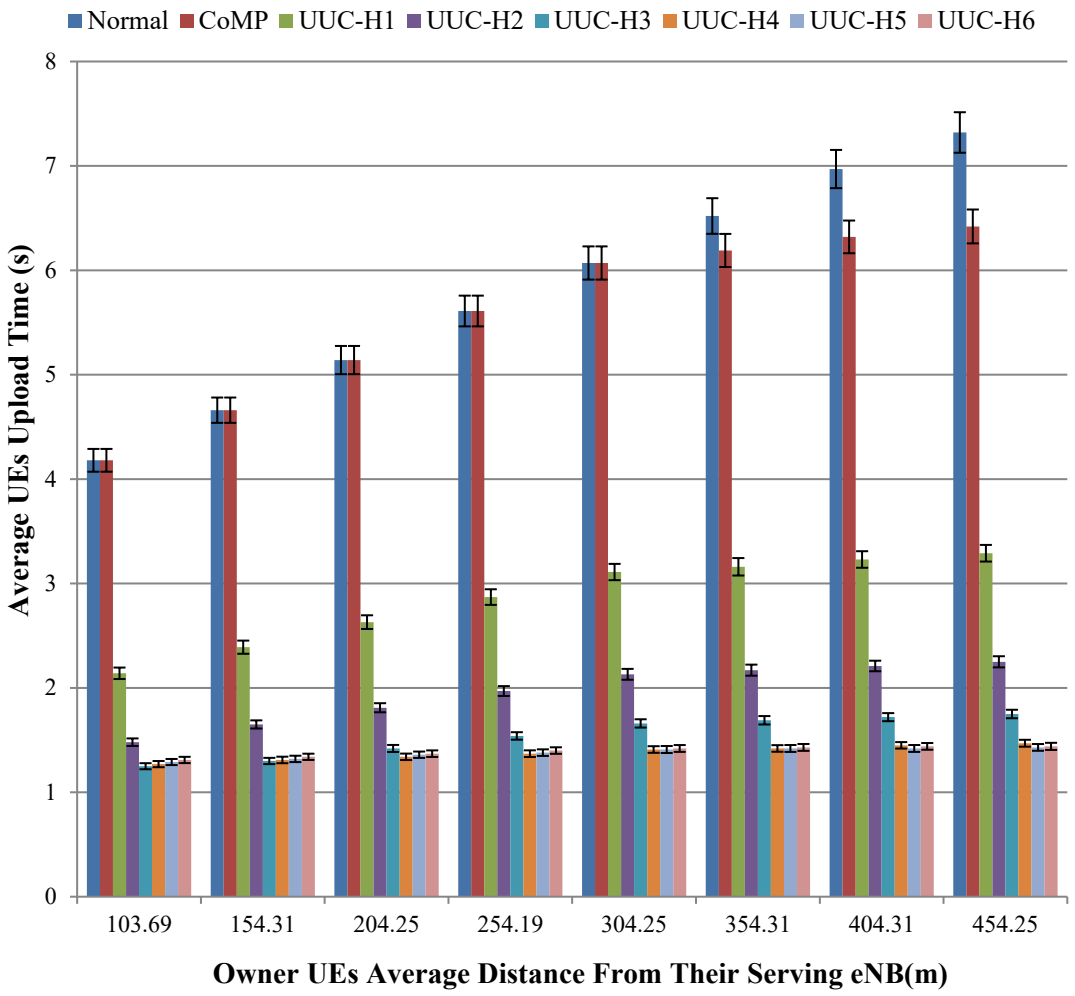


Figure 9.7: Average UEs upload time versus the UE-to- eNB distance

The UEs showed an interesting behavior when they used UUC. Close to the cell center, where the distance between the UEs and their serving eNB is about 103 m, adding a helper UE (UUC-H1) offered approximately a 48.5% reduction in the UEs' upload time compared to both the non-cooperative algorithm and CoMP. This number is about 74.5% and 70% for the UUC-H2 and UUC-H3 respectively. Actually, UUC-H3 provided the best performance among all the methods. The important point here is the fact that adding

more helper UEs (UUC-H4, UUC-H5 and UUC-H6) did not improve the users upload and it also reduced from the UUC performance gain compared to the UUC-H3. This happened because the overhead of adding more helper UEs outweighed the performance gain achieved by the addition of more helper UEs. However, as the distance between the UEs and their serving eNBs increases, adding more helper UEs seems to be promising. For example, when the UEs were about 204 or 404 m away from the cell center, UUC-H4 and UUC-H5 provided the best performance. UUC-H6 never provided the best performance. The latter means to make sure that we achieve to the best UUC performance, there should be a limit on the number of the helper UEs that can join the owner UE upload process. It also should be noted that in Figure 9.7 the entire UUC derivatives offered better upload performance compared to the non-cooperative algorithm and CoMP.

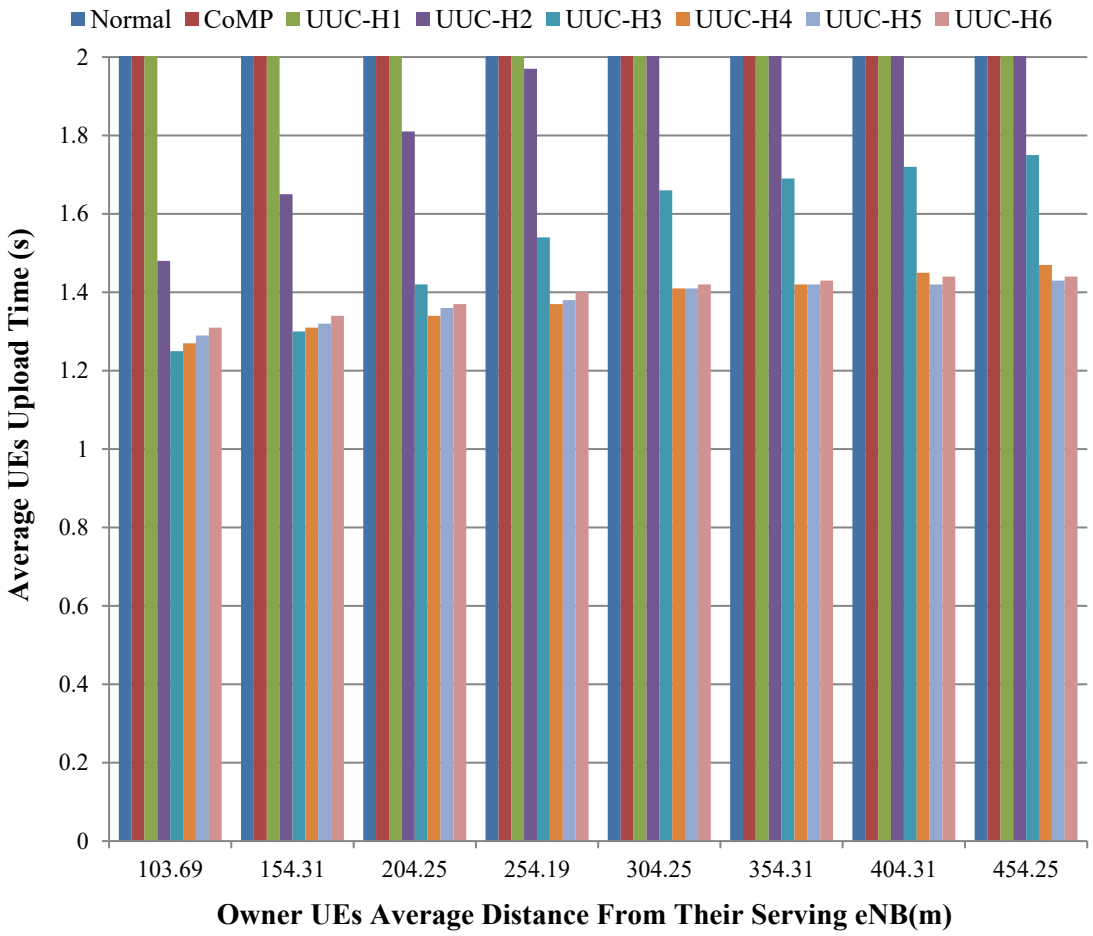


Figure 9.8: First two seconds of UEs upload time from Figure 9.7

Figure 9.9 shows the percentage of the data file pieces that the owner UE uploaded to the mobile network. The owner UE is completely responsible for the pieces upload when it employs the non-cooperative algorithm or CoMP. However, this is not the case when the owner UEs uses UUC. In this situation, the percentage of the data file pieces that the owner UE uploads is depend on the number of the helper UEs as well as the UEs distance from their serving eNBs. Figure 9.9 illustrates that as the latter increases, in the most cases this percentage decreases.

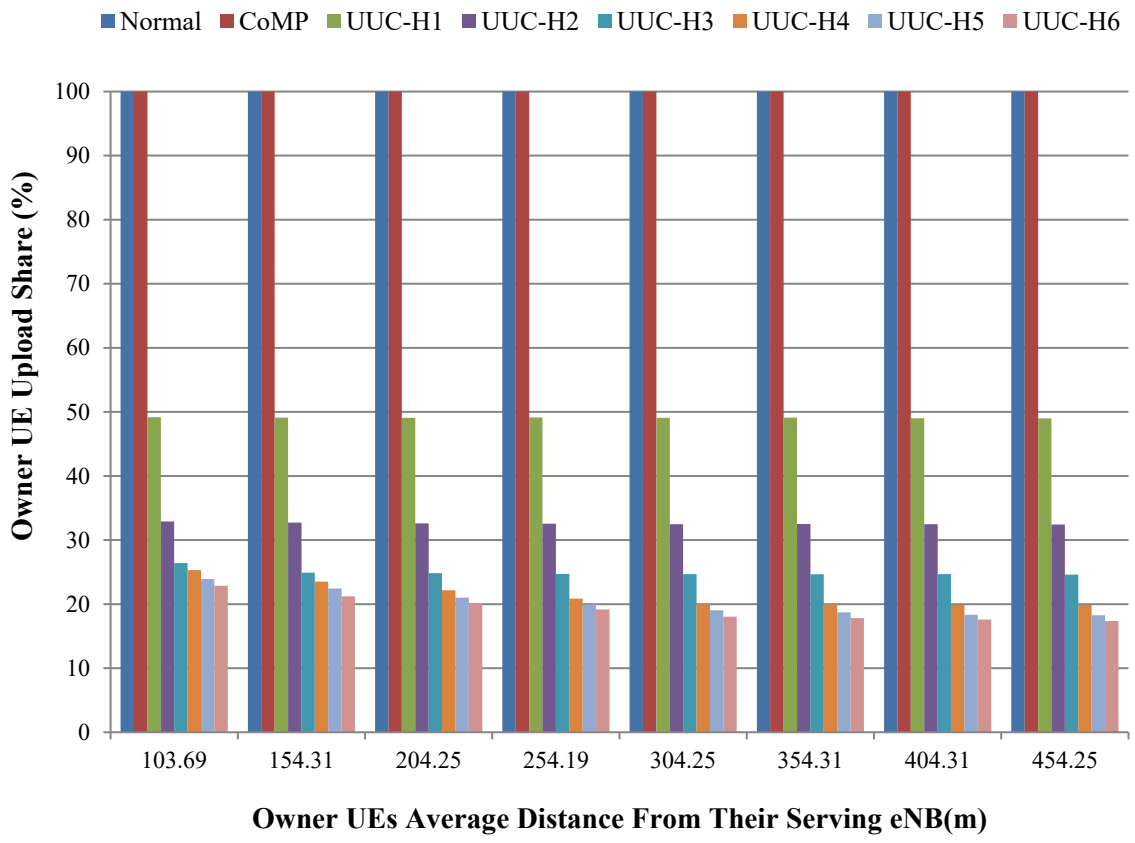


Figure 9.9: The percentage of the uploaded data files pieces by the owner UE

In the second set of simulations (scenario 2), we want to study the effect of the data file size on the UUC performance. Compared to the non-cooperative and CoMP, UUC imposes overhead to the mobile network. For example, in the UUC method, an owner UE and its helper UE require exchanging some control and data messages for the D2D communication channel setup and pieces transfer from the owner UE to the helper UE.

Therefore, this extra overhead on the D2D channels would reduce the upload performance of the owner UE while it deals with small data files. On the other hand, a relatively large file would make the effect of the UUC overhead, and it would make UUC a better option for the upload, since it uses multiple users to upload resources. In this scenario, we focus on cell-edge UEs that are located randomly between 350 m and 500 m from their serving eNBs. The owner UE to helper UE distance is kept fixed at 10 m.

Figure 9.10 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. Since, in scenario 2, we only changed the file size at each round of simulations, the average UEs distance from their serving eNBs remained fixed. The average number of eNBs that participated in the UEs uploads when they were using CoMP, UUC-H1, UUC-H2 or UUC-H6 is 2.26, 4.54, 6.81 and 15.93 respectively.

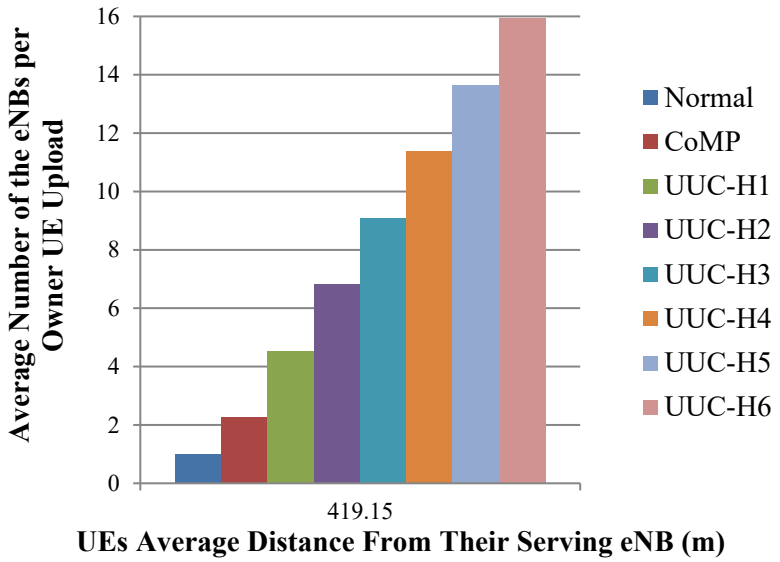


Figure 9.10: Average number of eNBs versus the UE-to- eNB distance

Figure 9.11 shows the average upload time for a data file as a function of file size for UUC, the non-cooperative algorithm and CoMP in an urban area setting. Figure 9.12 is a focused version of the Figure 9.11. It concentrates on the first ten seconds of the upload process of all the methods, and provides clearer picture of the difference among the upload time of different approaches. As seen in Figure 9.11 and Figure 9.12, for a small

file size (10 MB), the best performance is for UUC-H3. This approach offered about 77% and 63% reduction in the owner UEs upload time compared to the non-cooperative algorithm and CoMP in order. However, when we increased the file size, the UUC approaches with more helper UEs provided better performance gain compared to the UUC-H3. Actually, for the file sizes bigger or equal to 30 MB, UUC-H5 and UUC-H6 provided the best results. It should be noted that for the file sizes in the Figure 9.11, if we want to achieve to the best performance of UUC, we should consider a boundary for the number of the helper UEs that participate in the owner UE upload process. In addition, in this figure, the entire UUC derivatives offered better upload performance compared to the non-cooperative algorithm and CoMP.

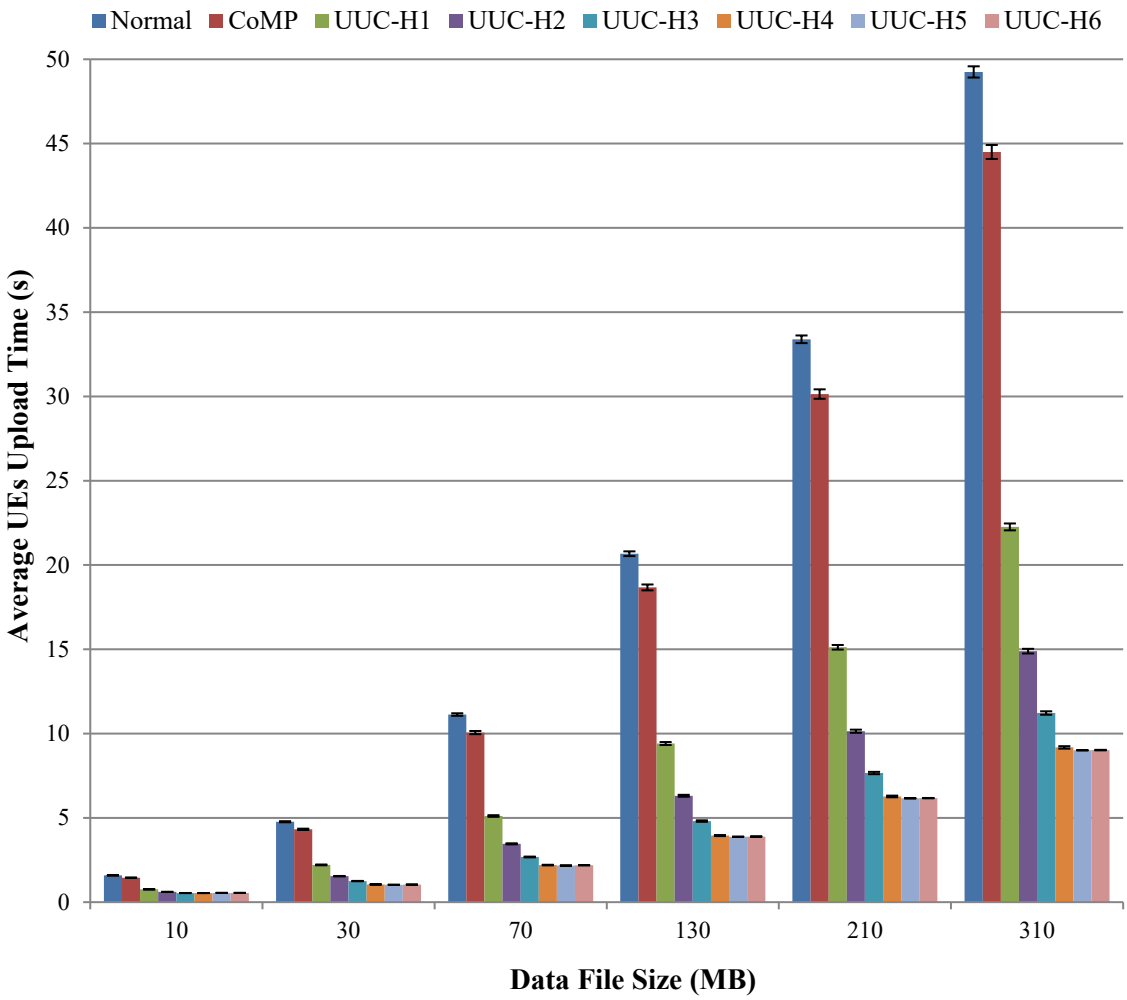


Figure 9.11: Average UEs upload time versus the UE-to- eNB distance

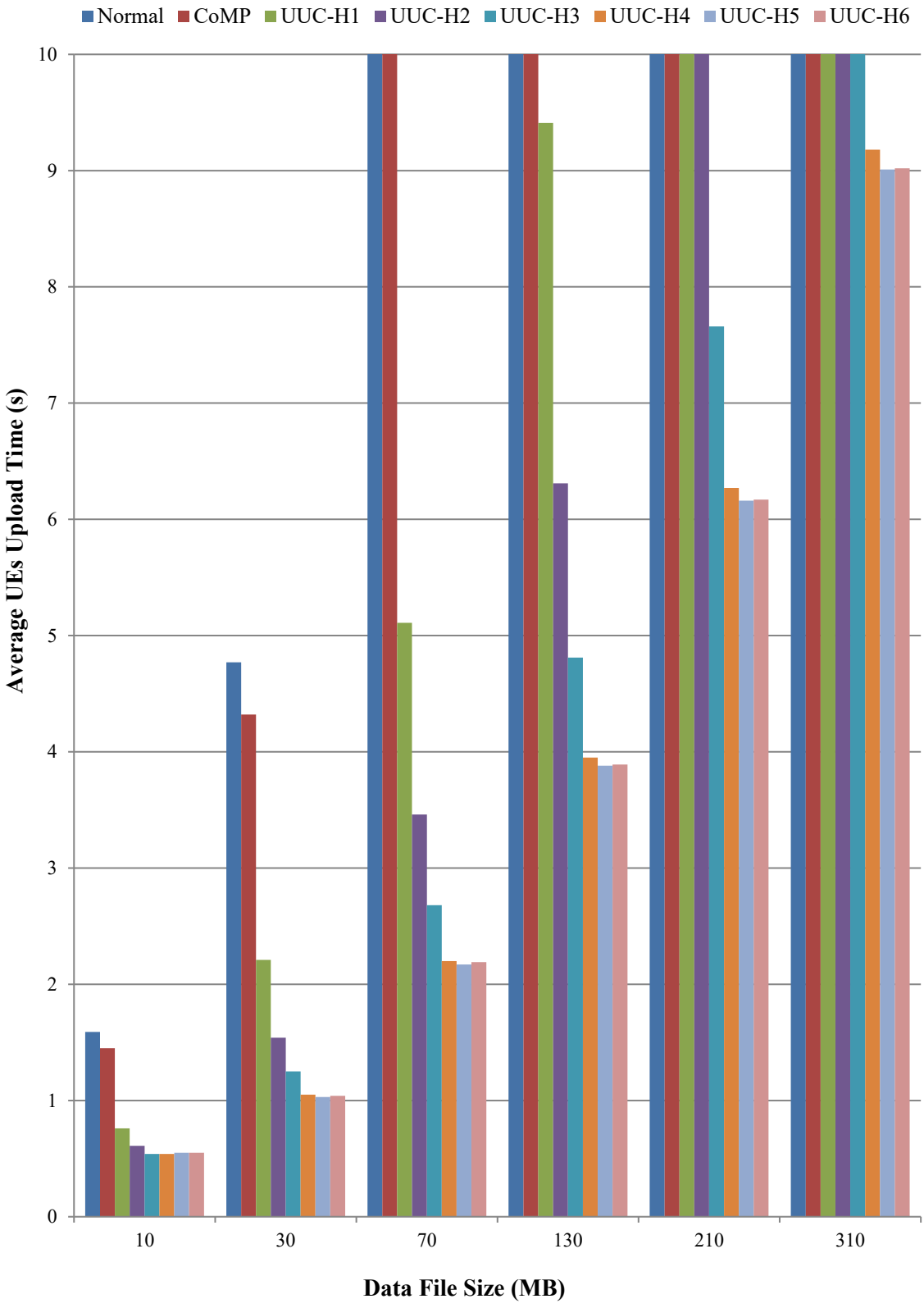


Figure 9.12: First ten seconds of UEs upload time from Figure 9.11

Figure 9.13 illustrates the percentage of the data file pieces that the owner UE uploaded to the mobile network in the second simulation scenario. Again, the owner UE uploads all the pieces when it employs the non-cooperative algorithm or CoMP as its upload algorithm. However, this is not the case when the owner UEs uses UUC. As seen in Figure 9.13, the owner UE uploads more pieces of the file when the file size is bigger than 10 MB. However, we could not see a general trend that is applicable to all the UUC approaches. For example, in UUC-H6, the percentage of the pieces that owner UE uploaded them has a gradual increase as the file size increases, but in UUC-H2, we observed that (in Figure 9.13) as the file size increased the percentage of the pieces that owner UE uploaded experienced different trends (almost it increased and decreased alternatively). However, this difference is too small and it can be because of the dynamic and random nature of the upload process in the mobile network.

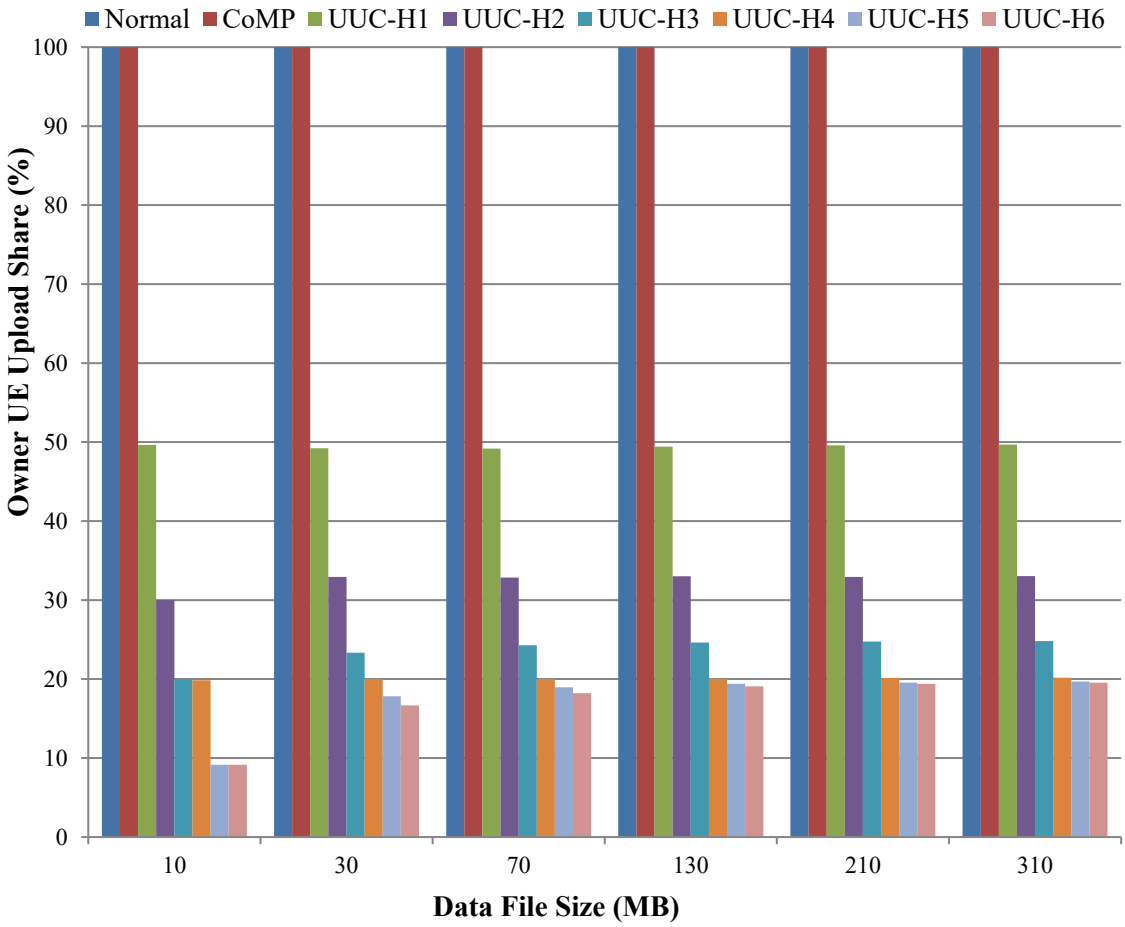


Figure 9.13: The percentage of the uploaded data files pieces by the owner UE

9.3 Model Verification and Validation

Similar to what we described in 8.3, we developed the UUC DEVS models in collaboration with experts from Ericsson Canada. In terms of model validation, the experts helped in checking that the conceptual model along with the assumptions and simplifications were correct. In term of model verification, we followed a same process as described earlier. We tested our DEVS model using a bottom-up approach as described earlier.

9.4 Summary

In this chapter, we discussed a DEVS model for LTE networks that was used to simulate and compare UUC to the non-cooperative method, as well as CoMP. We discussed the details of the coupled and atomic models that formed the UUC DEVS model together. We investigated the structure of these coupled models. We also presented a simplified UML class diagram for the atomic model classes. Moreover, we elaborated the list of the parameters and their values that we used in the simulations. Finally, we presented the simulation scenarios and their result. The simulation results show that UUC improves the UEs' upload performance regardless of their position within the cell, compared to the non-cooperative and CoMP.

10 Conclusion

Since the advent of mobile networks at the end of 1970s, they became very popular. These kind if wireless networks grew in such a way that at the time being (2016), the number of the mobile networks subscriptions is almost equal to the world population. Beside all its benefits, this enormous number of users introduces new challenges in the mobile network as well. The mobile network operators need to address these challenges, and provide suitable services for their customers to be able to keep them in this highly competitive market.

In recent years, another phenomenon, smart phones, changed the mobile network users' attitude considerably, and it raised serious challenges for the mobile network operators as well. Nowadays, people use these devices to perform many of the tasks that they used to do with other devices like cell phones, desktops, tablets, etc. Some of those tasks require high data rate services, and users expect the mobile network operators provide them such a service. This high data rate demand of users has an ever-increasing nature, and the mobile network operators require dealing with this high data rate demand as well.

Other important issue that mobile network operators need to take into account is the high-level requirements that international organizations such as 3GPP defined for the future mobile networks standards. Usually these requirements demand high-level quality of services for the customers, and the mobile network operators require employing advanced techniques to enhance their systems to meet these expectations. However, this is not an easy task. For example, according to these kinds of requirements the mobile network operators should provide high data rate services for their users regardless of their position in the covered area. It is even more complex when the UE is close to the cell edge. In this case, besides the lower signal strength, because of the distance between the UE and the eNB, the interference level from the neighboring eNBs is higher as the UE will be closer to them. Therefore, cell-edge UEs performance is another serious challenge for the

mobile network operators. We addressed these problems by proposing advanced algorithms to enhance the UEs performance.

Another goal of this thesis was to evaluate the performance of the proposed methods. To do so, we compared them with other methods under various simulation scenarios using both rural and urban area settings.

The Shared Segmented Upload (SSU) algorithm was one of the proposed solutions to enhance the cell-edge users upload in a LTE-Advanced mobile network with a distributed CoMP architecture. This method improves the cell edge user's uplink performance by transferring large files in the small segments from a single UE to the eNBs in a coordination set. This technique is a subset of the Joint Processing method in CoMP.

Another method that was proposed in this research was the Upload User Collaboration (UUC). This algorithm also improves the cell-edge users upload performance. To do so, UUC uses multiple UEs to enhance the upload process of a single UE. In the upload process, each UE divides its data file into a number of pieces. The UE that holds the data file starts the upload process by transferring pieces to the group of eNBs that are coordinated dynamically.

Both the SSU and UUC can be used in the current mobile network components and user devices. These methods do not require a specific hardware to be useable in the current mobile networks. However, some software updates in those devices is required to let them use the SSU and UUC methods. Considering the LTE-Advance as the target protocol of the mobile network, there should be some updates or modifications in different layers of the LTE-Advanced protocol stack.

The Super Set method was proposed to extend the UUC algorithm. The Super Set concept allows the owner and helper UEs to communicate with different set of eNBs as their coordination set. This method was proposed to solve the fixed coordination set problem for the UUC algorithm and any other algorithm, in which multiple UEs

cooperate in the upload process.

The handover process in the methods such as UUC, in which multiple UEs are cooperating in the upload process, is not as simple as the conventional handover process. Therefore, to improve the UUC method, and to allow both the owner and helper UE change their host cell, without losing their unfinished upload process, a handover method was developed. Likewise, a retransmission technique was proposed to improve the traditional retransmission procedure of the LTE-Advanced protocol stack.

Different DEVS models were developed for modeling and simulation of SSU. The simulation results showed that compared to the non-cooperative method, SSU provides services that are more consistent to the users as their distance increases from the cell center. In addition, SSU provided higher data rate for the users and reduced the time to upload its data from a UE to the network. Another DEVS model was presented for LTE networks, and it was used to simulate and compare the UUC algorithm to a non-cooperative method, as well as CoMP. 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). This model was also improved to support some of the super set models.

10.1 Future Work

Considering the large amount of data that required to be transmitted over a mobile network, further investigation is required to study its influence on the backhaul. Particularly, we need to investigate the SSU overhead on the backhaul in more detail. Moreover, we need to extend the proposed SSU algorithm to reduce such overhead. In addition, both the SSU and UUC methods introduce new messages that impose overhead to the system and this required to be studied in more detail. It should be mentioned that according to the simulation results (from chapters 8 and 9), these two methods (SSU and UUC) improve the cell edge UEs' performance considerably (specially the UUC method). It seems that the achieved gain from these two methods outweighed the overhead of the control messages (considering the small size of the control messages

versus the size of the data that need to be uploaded).

In addition, further investigation is necessary on the influence of piece sizes on the overhead of both the SSU and UUC algorithms on the backhaul. The goal in this direction can be developing a method that offers an optimal piece sizes to the UEs at the beginning of the upload process.

Moreover, we need to study the effectiveness of these algorithms when we are dealing with the upload of the delay-sensitive data. In particular, live audio-video systems as the soft real time systems can be the main targets of this study.

Extending the current UUC model to support the Super Sets as well as the Handover can be another interesting direction. The extended model can be employed to investigate the effectiveness of the UUC method, or any other method that uses multiple UEs in the upload process, when the UEs are scattered in the covered area. Likewise, a simulation model to support both the LTE-A standard retransmission process and the proposed retransmission method in this research are needed to evaluate the effectiveness of the proposed methods as well as its overhead on the mobile network backhaul.

The UUC algorithm showed its efficiency in both the CoMP and non-CoMP scenarios. However, we think that this method could be used for the downlink as well. The current model can be extended to support users download, and then it is possible to study the performance gain of the UUC when it is employed as a download schema.

Finally, using unlicensed bandwidth for the future mobile network communication became a hot research topic for the researchers as well as the mobile network operators. Considering the nature of this proposal, UUC could be a promising algorithm for both the uplink and downlink purposes in such a mobile network. This can be an attractive direction to be followed.

References

[1] Miniwatts Marketing Group, "Internet world stats, usage and population statistics." [Online]. Available: http://www.internetworldstats.com/emarketing.htm. [Accessed: 20-Oct-2016].

[2] Statistic Brain, "Internet Statistics." [Online]. Available: http://www.statisticbrain.com/internet-statistics/. [Accessed: 20-Oct-2016].

[3] CTIA The Wireless Association, "Annual Wireless Industry Survey." [Online]. Available: http://www.ctia.org/your-wireless-life/how-wireless-works/annual-wireless-industry-survey. [Accessed: 20-Oct-2016].

[4] Ericsson Mobility Report, "On the Pulse of the Networked Society," 2016.

[5] United Nation Population Division, "World Population Prospects The 2015 Revision," New York. USA, 2015.

[6] The Statistics Portal, "Percentage of all global web pages served to mobile phones from 2009 to 2016." [Online]. Available: https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/. [Accessed: 01-Nov-2016].

[7] M. Tavanpour, M. Moallemi, G. A. Wainer, J. Mikhail, G. Boudreau, and R. Casselman, "Shared Segmented Upload in Mobile Networks using Coordinated Multipoint," Provisional Patent Filing Reference P43130 US1, PCT/IB2015/051404, 2015.

[8] M. Tavanpour, G. A. Wainer, J. Mikhail, G. Boudreau, H. SeyyedmMahdi, and R. Casselman, "File Transfer by Mobile User Collaboration," Provisional Patent Filing Reference P46444 WO1, PCT/IB2015/054524, 2015.

[9] M. Tavanpour, G. Boudreau, and G. A. Wainer, "File Transfer by Mobile User Collaboration: Super Sets and Handover," Provisional Patent Filing Reference P48914 WO1, PCT/IB2016/051382, 2016.

[10] B. Cohen, "The BitTorrent Protocol Specification." [Online]. Available: http://bittorrent.org/beps/bep_0003.html. [Accessed: 09-Nov-2016].

[11] D. Seo, Evolution and Standardization of Mobile Communications Technology. IGI Global, 2013.

[12] S. Pearson, "The 1G (First Generation) Mobile Communication Technology Standards." [Online]. Available: http://network-communications.blogspot.ca/2011/07/1g-1st-generation-mobile-communications.html. [Accessed: 09-Nov-2016].

[13] D. Flore and B. Bertenyi, "Tentative 3GPP timeline for 5G." [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1674-timeline_5g. [Accessed: 09-Nov-2016].

[14] 2gprod, "Advantages Of 2G Technology." [Online]. Available: http://www.2gprod.com/advantages-2g-technology-review.html. [Accessed: 01-Feb-2016].

[15] I. Poole, "GSM: Global System for Mobile Communications Tutorial." [Online]. Available: http://www.radio-electronics.com/info/cellulartelecomms/gsm_technical/gsm_introduction.php. [Accessed: 01-Jul-2015].

[16] M. Rouse, "General Packet Radio Service." [Online]. Available:

http://searchmobilecomputing.techtarget.com/definition/GPRS. [Accessed: 01-Jul-2015].
[17] Tutorialspoint, "GPRS - Architecture." [Online]. Available: http://www.tutorialspoint.com/gprs/gprs_architecture.htm. [Accessed: 01-Oct-2015].
[18] I. Poole, "GSM EDGE tutorial." [Online]. Available: http://www.radio-electronics.com/info/cellulartelecomms/gsm-edge/basics-tutorial-technology.php. [Accessed: 01-Oct-2015].
[19] A. Osseiran, J. F. Monserrat, and W. Mohr, Mobile and wireless communications for IMT-advanced and beyond. John Wiley & Sons, 2011.
[20] ITU, "About ITU." [Online]. Available: http://www.itu.int/en/about/Pages/default.aspx. [Accessed: 01-Sep-2015].
[21] Telecom ABC, "IMT-2000." [Online]. Available: http://www.telecomabc.com/i/imt-2000.html. [Accessed: 01-Sep-2015].
[22] 3GPP, "About 3GPP." [Online]. Available: http://www.3gpp.org/about-3gpp. [Accessed: 01-Sep-2015].
[23] ICT Data and Statistics Division, "The world in 2015," Geneva, Switzerland, 2015.
[24] European Telecommunications Standards Institute, "Universal Mobile Telecommunication System (UMTS); General UMTS Architecture," 1999.
[25] European Telecommunications Standards Institute, "Universal Mobile Telecommunication System (UMTS); UTRAN Overall Descriptopn," 2001.
[26] I. Poole, "3G UMTS / WCDMA Network Architecture." [Online]. Available: http://www.radio-electronics.com/info/cellulartelecomms/umts/umts-wcdma-network-architecture.php. [Accessed: 01-Oct-2015].
[27] I. Poole, "HSPA – High Speed Packet Access Tutorial." [Online]. Available: http://www.radio-electronics.com/info/cellulartelecomms/3g-hspa/umts-high-speed-packet-access-tutorial.php. [Accessed: 01-Aug-2015].
[28] 3GPP, "The Evolved Packet Core." [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core. [Accessed: 01-Aug-2015].
[29] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description;," 2010.
[30] 3GPP, "NAS." [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/96-nas. [Accessed: 01-Oct-2015].
[31] LteWorld, "Non Access Stratum (NAS) Protocol." [Online]. Available: http://lteworld.org/specification/non-access-stratum-nas-protocol. [Accessed: 01-Oct-2015].
[32] LteWorld, "LTE Protocols and Specifications." [Online]. Available: http://lteworld.org/lte-protocols-specifications. [Accessed: 01-Oct-2015].
[33] Matthew Baker and Alcatel-Lucent, "LTE-Advanced Physical Layer," Beijing, Chaina, 2009.
[34] 3GPP, "LTE; Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN), 3GPP TR 25.913 version 8.0.0 Release 8.," 2009.
[35] E. Dahlman, S. Parkvall, and J. Skold, 4G: LTE/LTE-advanced for mobile

broadband. Academic press, 2013.
[36] P. Cerwall and Ericsson Mobility Report, "On the Pulse of the Networked Society," 2015.
[37] GSMA Head Office, "The Mobile Economy 2016," London, United Kingdom, 2016.
[38] WiMAX Forum, "Wimax." [Online]. Available: http://wimaxforum.org/TechSpec. [Accessed: 01-Oct-2015].
[39] 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.
[40] 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.
[41] S. Parkvall and D. Astely, "The evolution of LTE towards IMT-advanced," J. Commun., vol. 4, no. 3, pp. 146–154, 2009.
[42] T.-T. Tran, Y. Shin, and O.-S. Shin, "Overview of enabling technologies for 3GPP LTE-advanced," EURASIP J. Wirel. Commun. Netw., vol. 2012, no. 1, p. 1, 2012.
[43] V. Jungnickel et al., "Field trials using coordinated multi-point transmission in the downlink," in Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium on, 2010, pp. 440–445.
[44] R. Irmer et al., "Coordinated multipoint: Concepts, performance, and field trial results," IEEE Commun. Mag., vol. 49, no. 2, pp. 102–111, 2011.
[45] X. Zhang, M. A. Khojastepour, K. Sundaresan, S. Rangarajan, and K. G. Shin, "Exploiting interference locality in coordinated multi-point transmission systems," in 2012 IEEE International Conference on Communications (ICC), 2012, pp. 4207–4211.
[46] S. Mehryar, A. Chowdhery, and W. Yu, "Dynamic cooperation link selection for network MIMO systems with limited backhaul capacity," in 2012 IEEE International Conference on Communications (ICC), 2012, pp. 4410–4415.
[47] S. Brueck, L. Zhao, J. Giese, and M. A. Amin, "Centralized scheduling for joint transmission coordinated multi-point in LTE-Advanced," in Smart Antennas (WSA), 2010 International ITG Workshop on, 2010, pp. 177–184.
[48] D. Lee et al., "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," IEEE Commun. Mag., vol. 50, no. 2, pp. 148–155, 2012.
[49] X. Lu, W. Zhou, S. Chen, and J. Song, "Analysis of signaling design in LTE-Advanced coordinated multipoint transmission/reception system," in Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on, 2012, vol. 3, pp. 756–759.
[50] 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.
[51] V. Jungnickel et al., "Coordinated multipoint trials in the downlink," in Globecom Workshops, 2009, pp. 1–7.

[52] F. Malandrino, Z. Limani, C. Casetti, and C.-F. Chiasserini, “Interference-aware downlink and uplink resource allocation in HetNets with D2D support,” IEEE Trans. Wirel. Commun., vol. 14, no. 5, pp. 2729–2741, 2015.
[53] M. Erol-Kantarci, “Content caching in small cells with optimized uplink and caching power,” in 2015 IEEE Wireless Communications and Networking Conference (WCNC), 2015, pp. 2173–2178.
[54] 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.
[55] 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.
[56] 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.
[57] G. D. Boudreau, R. Casselman, and S. H. Seyedmehdi, “Mobile Device Assisted Coordinated Multipoint Transmission and Reception.” Google Patents, 2015.
[58] Q. Zhu, Q. Tang, Y. Yang, S. Xu, Y. Hu, and F. Liu, “Uplink resource sharing of multiple D2D links underlaying cellular networks,” in Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on, 2016, pp. 365–369.
[59] J.-Y. Pan, K.-W. Chang, and H.-W. Lin, “Uplink Shared Resource Scheduling for Device-to-Device Underlay Communication with Least Transmission Power Resource Selection and Guaranteed Link Quality,” in Networking and Network Applications (NaNA), 2016 International Conference on, 2016, pp. 364–369.
[60] A. Ajami and H. A. Artail, “A two stage PAPR reduction technique for the uplink of LTE-Advanced with carrier aggregation,” in Wireless Communications and Networking Conference (WCNC), 2016 IEEE, 2016, pp. 1–6.
[61] S. Hamda, M. Pischella, D. Roviras, and R. Bouallegue, “Cooperative uplink OFDMA-MIMO resource allocation with multiplexing relays,” in Wireless Communications and Networking Conference (WCNC), 2016 IEEE, 2016, pp. 1–7.
[62] M. A. Lema, E. Pardo, O. Galinina, S. Andreev, and M. Dohler, “Flexible Dual-Connectivity Spectrum Aggregation for Decoupled Uplink and Downlink Access in 5G Heterogeneous Systems,” IEEE J. Sel. Areas Commun., vol. 34, no. 11, pp. 2851–2865, 2016.
[63] D. Kong, J. Zeng, X. Su, L. Rong, and X. Xu, “Multiuser detection algorithm for PDMA uplink system based on SIC and MMSE,” in Communications in China (ICCC), 2016 IEEE/CIC International Conference on, 2016, pp. 1–5.
[64] Z. Yang, J. Cui, X. Lei, Z. Ding, P. Fan, and D. Chen, “Impact of Factor Graph on Average Sum Rate for Uplink Sparse Code Multiple Access Systems,” IEEE Access, vol. 4, pp. 6585–6590, 2016.
[65] V. Miliotis, L. Alonso, and C. Verikoukis, “Combating selfish misbehavior with reputation based uplink offloading for IP Flow Mobility,” in 2015 IEEE

International Conference on Communications (ICC), 2015, pp. 7012–7017.
[66] V. Miliotis, L. Alonso, and C. Verikoukis, "Energy efficient proportionally fair uplink offloading for IP Flow Mobility," in 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014, pp. 6–10.
[67] V. Miliotis, L. Alonso, and C. Verikoukis, "Offloading with IFOM: The uplink case," in 2014 IEEE Global Communications Conference, 2014, pp. 2661–2666.
[68] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in INFOCOM, 2011 Proceedings IEEE, 2011, pp. 882–890.
[69] F. Malandrino, C. Casetti, C. F. Chiasserini, and Z. Limani, "Uplink and downlink resource allocation in D2D-enabled heterogeneous networks," in Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE, 2014, pp. 87–92.
[70] L.-H. Hsu, H.-L. Chao, C.-L. Liu, and K.-L. Huang, "Multi-user MIMO scheduling in LTE-advanced uplink systems," in 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2013, pp. 1811–1816.
[71] S. Mumtaz, K. M. S. Huq, and J. Rodriguez, "Coordinated paradigm for D2D communications," in Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, 2014, pp. 718–723.
[72] Y. Li, J. Zhang, X. Gan, L. Fu, H. Yu, and X. Wang, "A Contract-Based Incentive Mechanism for Traffic Offloading over Delay Tolerant Networks," IEEE Trans. Wirel. Commun., vol. 15, no. 8, pp. 5314–5327, 2016.
[73] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Motivating smartphone collaboration in data acquisition and distributed computing," IEEE Trans. Mob. Comput., vol. 13, no. 10, pp. 2320–2333, 2014.
[74] G. A. Wainer, M. Tavanpour, and E. Broutin, "Application of the DEVS and Cell-DEVS formalisms for modeling networking applications," in Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013, 2013.
[75] G. A. Wainer, Discrete-Event Modeling and Simulation: A Practitioner's Approach. CRC Press, 2009.
[76] 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.
[77] 3GPP TR 36.942 version 10.2.0, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency (RF) system scenarios, (3GPP TR 36.942 version 10.2.0 Release 10)," 2011.
[78] 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.
[79] 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.
[80] 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.
[81] X. Zhang and X. Zhou, LTE-advanced air interface technology. CRC Press, 2012.
[82] 3GPP TR 36.843 version12.0.0, "Study on LTE device to device proximity services; Radio aspects," 2014.
[83] J. Meinilä, P. Kyösti, and others, "D5. 3: WINNER+ Final Channel Models V1. 0," ht tp//projects. Celt. org/winner+/WINNER+% 20Deliverables D, vol. 5, 2010.
[84] J. Meinilä, P. Kyösti, and Others, "Channel Models, D1. 1.2 V1. 2," 2008.