

OPTIMIZATION OF RADIO AND INFRASTRUCTURE RESOURCES FOR EFFICIENT MASSIVE DATA TRANSMISSION AND STORAGE IN MOBILE CLOUD OVER 5G HETNET

by

Richa Siddavaatam

Master of Applied Science, Ryerson University, 2012

Bachelor of Engineering, Pune University, 2005

A dissertation

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

in the Program of

Computer Science

Toronto, Ontario, Canada, 2019

© Richa Siddavaatam 2019

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A DISSERTATION

I hereby declare that I am the sole author of this dissertation. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

OPTIMIZATION OF RADIO AND INFRASTRUCTURE RESOURCES FOR EFFICIENT MASSIVE
DATA TRANSMISSION AND STORAGE IN MOBILE CLOUD OVER 5G HETNET

Doctor of Philosophy 2019

Richa Siddavaatam

Computer Science

Ryerson University

Abstract

In a dense HetNet enhanced with MCC and MEC technologies, an efficient usage of limited radio resources and augmented network infrastructure is critical to achieve the best possible user experience and network performance, while realizing an efficient and effective response to the big data requirements of mobile devices. Considering this 5G-based dense HetNet architecture, this thesis proposes: (1) a Redundant Array of Independent Disks (RAID) based data storage algorithm that minimizes the data transmission delay during transmission of a big data file from the UE to the cloud data centre; (2) a mechanism design based algorithm to efficiently transfer big data files with minimum transmission delay from the UE to the cloud data centre; and (3) an Ant Colony Optimization (ACO) based energy-aware algorithm to jointly allocate the radio resources and transmit the big data file from the UE to the cloud data centre. Simulation results are provided, demonstrating the effectiveness of the proposed algorithms in comparison to the Greedy and the Data Split Multiple UE (DSMU) algorithms in terms of transfer delay, time required by a UE to store the big data file, and energy consumed by the UE during transmission.

Acknowledgment

I am indebted to my supervisor, Dr. Isaac Woungang who believed in my research abilities and gave me endless support. I am thankful to him for offering me such a great opportunity to join the DABNEL lab for conducting research in wireless communications. Under his supervision, I learned how to analyze research problem and find optimal solutions to it. Dr. Woungang inspired me by his hardworking and passionate attitude. I am grateful to him for all the guidance provided despite the challenges he faced on his personal front.

Also, I would like to thank my entire thesis committee for all insightful and valuable feedback provided.

As a proud Canadian, I would like to express a deep gratitude towards this beautiful country for providing me this amazing life and offering a lot of opportunities and providing success within my reach; all I did is to grab it. My special thanks goes to the fantastic day care staff at Woodbine Early Learning and Child Care Center who provided excellent care for my kids while I was busy with my research activities.

I can never find the words to thank my husband Prathap who has been my best friend, philosopher, guide. Without his support, encouragement and unconditional love, I would not be where I am today. He not only believed in me and was always supportive but he also offered me his advice and his valuable help.

I shall be eternally grateful to my son Aarav, daughter Reeva and unborn son Reyansh for being source of happiness in my life.

I dedicate this Ph.D dissertation to my better half without whom this research work wouldn't have been possible.

CONTENTS

1. Introduction	1
1.1 Thesis Motivation	1
1.2 Problem Statement	3
1.3 Methodologies	5
1.4 Research Contributions	6
1.5 Thesis Outline	9
2. Background and Literature Review	11
2.1 Background	12
2.1.1 The Need for Heterogeneous Wireless Networks	12
2.1.2 MCC and Mobile Edge Computing in a Cloud-Centric System	15
2.1.3 Intercloud over HetNet for MCC Applications in Emerging 5G Systems: A Resource Management Perspective	17
2.1.4 Network Simulator for 5G HetNet up-link	18
2.1.4.1 System Parameters for 5G Up-Link	19
2.1.4.2 Location of Network Elements	19
2.1.4.3 Cell Association for all UEs: Biased Cell Association Method	21
2.1.4.4 Channel Model	22
2.1.4.5 PRB Allocation to Up-Link Data Paths	24
2.1.4.6 Calculation of the SINR, Data Rate and Other Desired Network Performance	24
2.2 Literature Review	25

3.	Efficient Methods for Mobile Big Data Transfer Over HetNet in Emerging 5G Systems	32
3.1	Introduction	32
3.2	RAID Based Algorithm for Mobile Big Data Transfer Over HetNet in 5G Systems: Case of Equal Data Chunk Size	35
3.2.1	Calculation of the Bandwidth Available to Each UE	38
3.2.2	Lost Data Recovery Technique	39
3.3	Simulation Results: Case of Equal Data Chunk Size	40
3.4	RAID Based Algorithm for Mobile Big Data Transfer Over HetNet in 5G Systems: Case of Data Chunk Size Proportional to Uplink Capacity	44
3.4.1	Lost Data Recovery Technique	46
3.4.2	Simulation Results: Case of Data Chunk Size Proportional to Uplink Capacity	47
3.5	Application of the Proposed RAID Based Algorithm for Big Video Data Files Transfer Over HetNet in 5G Systems	49
3.5.1	Big Video data File Preprocessing	49
3.5.2	Proposed Big Video Data File Transmission Algorithm	50
3.5.3	Simulation Results: RAID Based Algorithm for Big Video Data Files Transfer	51
3.5.4	Potential Drawbacks of Proposed Approach	56
4.	Mobile Cloud Storage over 5G: A Mechanism Design Approach	57
4.1	Introduction	57
4.2	Preamble	60
4.2.1	Multiple Data Paths Available to a UE	60
4.2.2	Mechanism Design	61

4.3	Proposed RAID based Data Transmission Management Scheme	62
4.4	Proposed Radio Resource Management Scheme	63
4.4.1	PRB Allocation Problem	63
4.4.2	Proposed Algorithmic Solution to the PRB Allocation Problem	64
4.4.2.1	PRB Allocation Problem as GCP:	64
4.4.2.2	Proposed PRB Allocation and Pricing Algorithm:	65
4.4.2.3	Output Function:	69
4.4.2.4	Payment Function:	70
4.5	Analysis of the Proposed mechanism design based Method	71
4.5.1	Incentive Compatible in Ex-post Nash Equilibrium	71
4.5.2	Efficiency	73
4.5.3	Individually Rational	73
4.6	Simulation Results	74
4.6.1	Greedy Algorithm	74
4.6.2	Data Split Multiple UE Algorithm	75
4.6.3	Incentive Compatible ex-post Nash Equilibrium	79
5.	Joint Optimization of Radio and Infrastructure Resources for Energy Efficient Massive Data Storage in Mobile Cloud over 5G HetNet	83
5.1	Introduction	83
5.2	Optimization Problem	85
5.3	Proposed ACO based Approach	88
5.3.1	Ant Colony Optimization Paradigm	88
5.3.2	Ant Colony Optimization Metaheuristic	90
5.3.3	Proposed ACO Based Algorithm	91
5.3.4	Pheromone Definition	94

5.3.5	Lost Data Chunk Recovery Technique	95
5.4	Performance Evaluation of the Proposed ACO-Based Scheme . . .	96
5.4.1	Convergence of the Proposed ACO Algorithm	97
5.4.2	Simulation Results	98
5.5	Appendix for Chapter.5	104
5.5.1	Proof of energy consumption equation	104
6.	Conclusion	106
6.1	Conclusion	106
6.2	Future Work	108
	References	109
	Glossary	117

LIST OF FIGURES

1.1	Intercloud and HetNet for MCC applications	4
2.1	HetNet with three RATs and the associated random events	14
2.2	MEC over 5G systems	16
2.3	Overview of simulator	19
2.4	Location of network elements	20
3.1	System Model.	33
3.2	Data paths available between UE-1 and the cloud system in fig. 3.1.	33
3.3	Simulated location of the HetNet elements.	41
3.4	Data transfer delay vs. data file size.	42
3.5	Data transfer delay vs number of UEs.	43
3.6	Data transfer delay vs. number of femto BSs.	43
3.7	RAID-based algorithm for faster data transfer: Case of data chunk size proportional to uplink capacity	45
3.8	Data transfer delay vs. data file size	47
3.9	Data transfer delay vs number of UEs	48
3.10	Data transfer delay vs number of femto BSs	49
3.11	RAID-based algorithm for Big video data files transfer	52
3.12	Performance of video compression algorithm.	53
3.13	Impact of the variation in data file size on the data transmission delay.	54
3.14	Impact of the variation in the number of active UEs in the network on the data transmission delay.	55

3.15	Impact of variation in number of small BSs on the data transmission delay.	56
4.1	System Model.	58
4.2	Simulated network scenario.	66
4.3	Interference graph with $\gamma_{thresh} = 20\text{dB}$	66
4.4	Interference graph with $\gamma_{thresh} = -10\text{dB}$	67
4.5	Interference graph with $\gamma_{thresh} = -40\text{dB}$	67
4.6	Procedural details of the proposed solution comprising of DMLDR and RRM schemes.	69
4.7	Time required by the UE to complete the transfer of the file vs. size of data file (on a logarithmic scale).	76
4.8	Time required by the UE to complete the transfer of the file vs. number of active UEs in the network (on a logarithmic scale). . .	76
4.9	Time required by the UE to complete the transfer of the file vs. number of small cell BSs in the network.	77
4.10	Payment function distribution.	78
4.11	Initial number of PRBs assigned to a UE by the proposed algorithm for various declared type (t_i).	79
4.12	Final number of PRBs assigned to a UE by the proposed algorithm for various declared type (t_i) using frequency reuse.	80
4.13	Delay experienced by the UE and the payment value a UE needs to pay for various values declared type (t_i) declared.	81
4.14	Net utility of a UE for various values of type (t_i) declared.	81
5.1	System Model.	84
5.2	Proposed ACO-based solution made of a data transmission strategy and a PRB allocation and a loss data recovery scheme.	89
5.3	Travel map followed by an ant.	91
5.4	ACO based algorithm for data transmission strategy and PRB allocation	93

5.5	Simulated instance of network scenario.	96
5.6	Convergence of the proposed algorithm.	97
5.7	Impact of data transmission delay on the data file size	98
5.8	Data transmission delay when the number of UEs in the network is increased	99
5.9	Impact of data transmission delay when the number of small cell BSs in the network is increased	100
5.10	Energy dissipated by a UE during transmission when the data file size is increased	101
5.11	Energy dissipated by a UE during transmission of the data file when the number of UEs in the network is increased	102
5.12	Performance of proposed and other comparison algorithms to vary- ing number of small cell BSs in the network.	103

Chapter 1

Introduction

1.1 Thesis Motivation

With the advent of mobile smart communication devices and Internet of Things (IoT) as a new computing and communication paradigm [1], the emerging fifth generation (5G) network of wireless systems will feature an ever-growing demand for communication and computational requirements of modern mobile applications and services. To meet these requirements, Mobile Network Operators (MNO) are encouraged to explore the benefits of two complementary technologies that have received an increased momentum in the recent years, namely heterogeneous wireless networks (HetNet) and Mobile Cloud Computing (MCC).

Typically, a HetNet is composed of several wireless networks with different characteristics, it has been recognized as a viable solution to enhance the network capacity in terms of spectral efficiency, coverage area, supported data rates, to name a few. On the other hand, MCC has been advocated as a viable technology to boost the user equipment (UE) capabilities such as computation resources, storage space, battery lifetime, to name a few, by transferring the storage and computationally intensive tasks of the UEs to the resource-rich cloud data centers. However, one of the critical challenges with the current MCC solutions is that there is an increased latency occurring over a standalone wireless network

deployment when massive data are to be offloaded in the cloud. This has led to consider latency awareness as a driving force behind the cloudification process faced by the MNOs. This process, which involves the operations of Base Stations (BS) or Access Points (AP) and virtualized edge servers, is realized by means of the Mobile Edge Computing (MEC) paradigm [2], considered as a step towards low latency cloud storage on top of the MCC pathway.

The success of MCC mobile cloud applications depends on the effectiveness in the realization of the MEC deployment [3], and this is tightly coupled to the allocation of computing and communications resources. Therefore, resource management has become one of the main aspects in the design of a MEC based network solution. In this regard, several joint communications and computation resource management solutions have been proposed in the literature [4]. However, from their applicability standpoint, the volume, velocity and variety of the big data deluge may cause a threat to the service provisioning in such a way that MNOs may not be able to guarantee the strict latency control requirement of storage hungry and computation hungry MCC applications. Thus, a solution must be sought to make a better use of the network resources while making optimal use of the network infrastructure to effectively support advanced MCC application [5]. In this sense, a synergistic exploitation of HetNet enhanced with MCC and MEC technologies and 5G should be put forward to support advanced resource management applications. This has led to the realization of the so called Inter-cloud over HetNet (Fig. 1.1), where several UEs are expected to take advantage of the resource-rich cloud service providers (CSPs) to offload their mobile cloud applications assuming that a coalition prevails between an MNO and the CSPs (via the Cloud Service Broker (CSB)).

1.2 Problem Statement

In the aforementioned Intercloud over HetNet, the access to MCC services is performed via the HetNet, which itself is composed of macro BS, macro cells and small cells with MEC servers. Upon the establishment of the connections with the Base Stations (BSs), a UEs service request is transferred (by the Common Radio Resource Manager (CRRM) [6] to the CSB who decides which of the CSPs is best suited to host and handle it. An Intercloud over HetNet can then be viewed as a coalition of a massive deployment of overlapping radio access technologies/networks (RATs) controlled by a logical entity (here the CRRM) and multiple CSPs sharing their individual cloud resources, themselves controlled by the CSB [6]. Considering this generic infrastructure (Fig.1.1), the promises of dynamic selection of the best possible small cells for efficient task offloading and the best possible CSP for efficient task computation purpose has become a reality [7], [8], as well as the promises of addressing successfully the increasing latency problem perceived by existing mobile computing applications over standalone deployments [9].

Complementary to the work done so far [4, 5, 7, 9, 10], and considering the network infrastructure shown in Fig.1.1, where the assumption that a coalition exists between the MNO and the CSPs that will back up the MNO when MEC servers are running out of space, this thesis proposes algorithmic solutions to two interconnected problems, namely:

1. **Problem 1:** How to design efficient methods for the transmission of big data files from mobile end users (UEs) to the cloud data centers;
2. **Problem 2:** How to design an efficient method that can jointly optimize the radio resources and infrastructure resources for energy efficient big data transmission in mobile cloud over 5G HetNet,

Our proposed methods to tackle Problem 1 and Problem 2 are designed by taking advantage of the several benefits that are offered by the small cell densi-

fication and the overlapping and collaborative nature of the HetNet deployment [8], which allow mobile users (UEs) to be associated with the best BS or AP while maintaining a high mobility profile.

In the proposed solutions, the radio and infrastructure resources are optimally exploited and allocated to multiple competing users in such a way as to maximize the overall system's performance.

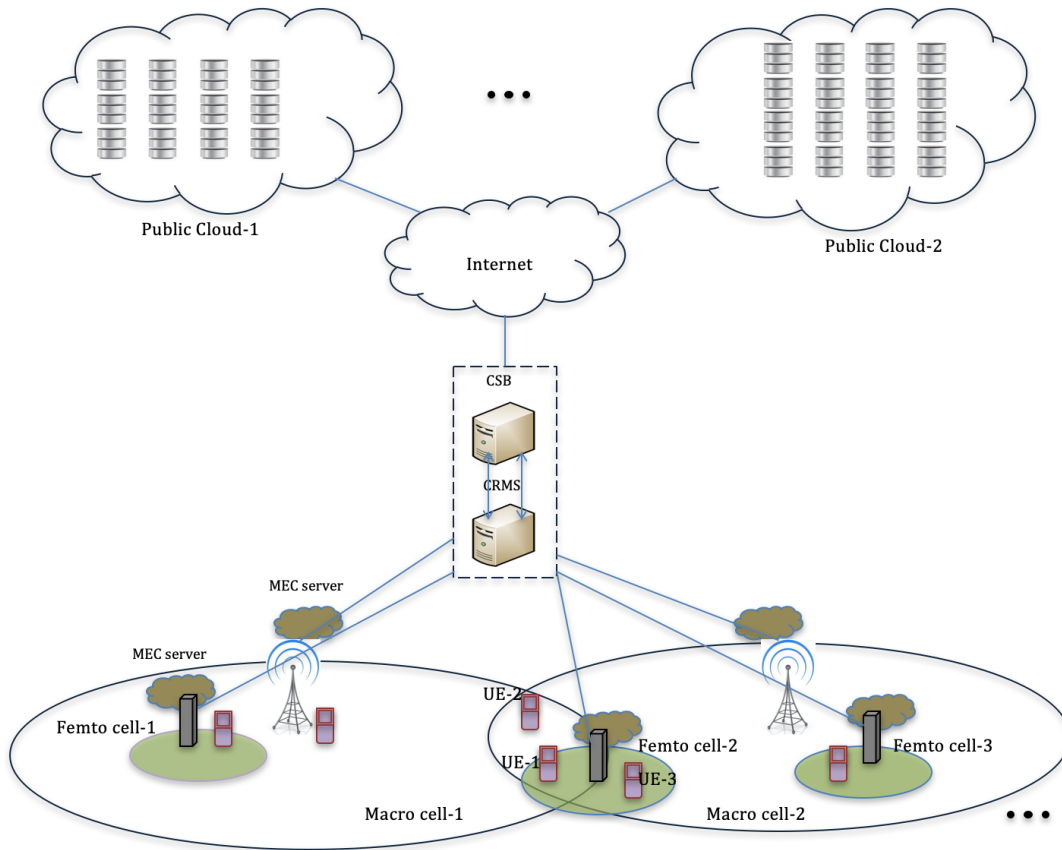


Figure 1.1: Intercloud and HetNet for MCC applications

1.3 Methodologies

To tackle Problem 1, the first approach (termed as Approach 1) consists of splitting the big data file into a number of chunks equal to the number of data paths available to the UE under consideration [11], which are then transmitted concurrently over these multiple data paths to the cloud. In this design, an instance of the system model depicted in Fig. 1.1 is considered, where the network consists of macro BSs, femto BSs with overlapping coverage areas, and where each BS has access to a public cloud system through a back-haul connection and there exists a MEC server associated with each BS.

The considered network serves a number of mobile devices randomly dispersed over the geographical area, and the BS association decision with a UE depends on the signal to interference and noise ratio (SINR) experienced by the user on the downlink transmission link (we assume reciprocal uplink-downlink channels), knowing that the UE can be associated to multiple BSs. In this setting, since the data chunks are transmitted on unreliable wireless data links, higher rate of failure to retrieve data chunk may occur. Thus, a data error correction technique is also proposed to address this deficiency, providing that the data file chunks and the lost data recovery syndromes be transmitted simultaneously on the above mentioned available data paths.

The proposed data transmission algorithm inspired from the Redundant Array of Inexpensive Disks-6 (RAID-6) technique [12], which exploits these data paths to expedite the data transmission process. The algorithmic solution derived from Approach 1 is also applied to the specific case of a big video file transmission [13], where the video file is compressed by using the H.265 compression algorithm [14] before getting transmitted over the network. Approach 1 is further modified by considering that the size data chunks are proportional to the uplink capacity of the data link on which it is to be transmitted [13].

The second approach to tackle Problem 1 (termed as Approach 2) is a mechanism design driven resource management approach [15]. In this approach, a dense

5G HetNet architecture enhanced with MEC storage centres is considered (as an instance of Fig 1). A multiple users scenario is also considered, where each user has multiple opportunities to simultaneous transfer its big data through multiple wireless links. The proposed mechanism design based approach consists of a data transmission management scheme, a physical resource block (PRB) allocation scheme and a pricing function, using the principle of algorithmic mechanism design [16]. Such design is meant to control the allocation of wireless resources in such a way as to prevent an abusive user from selfishly consume these resources.

The approach to tackle Problem 2 (termed as Approach 3) consists of formulating it as a combinatorial optimization problem, then solving it using an ACO based energy aware algorithm [17] involving the aforementioned data transmission management scheme, PRBs/bandwidth allocation scheme, and a DSMU algorithm.

Through simulation studies, the effect of different key parameters such as the delay taken by the UE to transfer the big data file (transmission delay) and energy consumed during the data transmission, on the performance of the proposed algorithms in comparison to that of some baseline algorithms, is evaluated.

1.4 Research Contributions

The main objective of this research is to design some methodologies for efficient transmission of big data files over HetNet in emerging 5G network systems, considering the network architecture depicted in Fig.1.1, where it is assumed that a coalition prevails between the MNO and CSPs, enabling the sharing of available resources from the CSB, HetNet, and CRMS realms, as well as the possibility to back up the MNO when MEC servers are running out of space.

The key contributions of this thesis are summarized as follows:

- Designing a RAID based data transmission algorithm that minimizes the data transmission delay during transmission of a big data file from the UE

to the cloud data center in a dense 5G network HetNet enhanced with MCC and MEC technologies (as shown in Fig.1.1).

- Develop a lost data recovery technique to deal with the problem of failure of a data chunk retrieval during transmission.
- Employ the cell range expansion (CRE) technique [18] (as biased cell association method) to increase the usability of small cell BSs.
- Apply the proposed algorithm to achieve an efficient transfer of mobile big data, in particular video data files, where in this case, the H.265 compression algorithm is invoked to compress the video file before transmitting it over the network from the UE to the cloud.
- Develop a network simulator for 5G HetNet uplink and use it to evaluate and discuss the effectiveness of the proposed algorithms in comparison of the conventional method, in terms of transfer delay and time required by a UE to store the file. In the conventional method, a big data file of the same size is transferred to the cloud storage centers via the macro BS with the highest SINR value using the back-haul connection.
- Developing a mechanism design based algorithm to efficiently transfer big data files with minimum transmission delay in a dense 5G HetNet enhanced with MCC and MEC technologies (as shown in Fig.1.1).
 - Formulate the radio resource (PRB) and infrastructure resource allocation problem as a mechanism design problem.
 - Develop an efficient heuristic to solve the formulated optimization problem, which consists of a PRB/ bandwidth allocation scheme, a pricing algorithm, and a data transmission management scheme involving the use of the above mentioned lost data recovery technique implemented at the individual UE level.

- Develop a Data Split Multiple UE (DSMU) Algorithm.
 - Implemented the proposed mechanism design based algorithm on the simulated 5G network model to evaluate and discuss its effectiveness in comparison to the Greedy and DSMU algorithms in terms of data transmission delay, the response of the algorithms to increased service load (i.e. an increase in the number of UEs) and the response of the algorithms to an enhancement in the network infrastructure (i.e. an increase in the number of small BSs).
 - Provide a mathematical analysis of the proposed algorithm to demonstrate its desirable properties from a game theoretical perspective such as incentive compatibility, efficiency and individual rationality.
- Developing a ACO based energy aware algorithm to jointly and effectively allocate the radio resources and transmit the big data file in a dense 5G HetNet enhanced with MCC and MEC technologies (as shown in Fig. 1.1).
 - Formulate the radio resource allocation and data transmission strategy as a multi-objective combinatorial optimization problem to minimize average data transmission delay experienced by network users and total energy consumption.
 - Develop an efficient ACO based heuristic to solve the formulated optimization problem, which consists of a modified version of the aforementioned PRB/bandwidth allocation scheme, data transmission management scheme, and lost data recovery technique.
 - Utilize the above mentioned network simulator for 5G HetNet uplink to evaluate and discuss the effectiveness of the proposed mechanism design based algorithm in comparison of the Greedy and DSMU algorithms in terms of data transmission delay, the response of the algorithms to increased service load (i.e. an increase in the number of UEs)

and the response of the algorithms to an enhancement in the network infrastructure (i.e. an increase in the number of small BSs).

1.5 Thesis Outline

The remainder of this thesis is organized as follows.

In Chapter 2, we present a brief overview of the considered dense 5G HetNet enhanced with MCC and MEC technologies (Fig.1.1). The most important features of such architecture from a resource management viewpoint such as the interoperation features between the Intercloud and HetNet, and the need of massive storage for mobile big data in such environment, are highlighted. We also present a literature review of the most representative resource management schemes to boost the storage and computational capabilities of mobile devices using MCC and HetNet technologies. Finally, the developed HetNet model simulator is described.

In Chapter 3, we describe the design of a RAID based data transmission algorithm that minimizes the data transmission delay during transmission of a big data file from UE to cloud data center in an MCC-HetNet environment. We also evaluate by simulations its effectiveness in terms of data transmission delay and the response of proposed algorithm to variation in service load and enhancement of network infrastructure compared to that of conventional method. In addition to this, a variation of above transmission scheme specifically designed for storage of big video files is presented in this Chapter.

In Chapter 4, we describe the design of a mechanism design based algorithm to efficiently transfer big data files with minimum transmission delay in a dense 5G HetNet enhanced with MCC and MEC technologies. We also evaluate by simulations its effectiveness in of transfer delay and the response of proposed algorithm to variation in service load and enhancement of network infrastructure compared to that of the Greedy and DSMU algorithms.

In Chapter 5, we describe the design of a ACO based energy aware algorithm

to jointly allocate the radio resources and transmit the big data file in a dense 5G HetNet enhanced with MCC and MEC technologies. We also evaluate by simulations its effectiveness in of transfer delay and the response of proposed algorithm to variation in service load and enhancement of network infrastructure compared to that of the Greedy and DSMU algorithms.

Finally, Chapter 6 concludes the thesis and presents some directions for future work.

Chapter 2

Background and Literature Review

The proliferation of devices, which are able to directly connect to the Internet, has led to a new computing and communication paradigm known as Internet of Things (IoT) [1]. In IoT, things are expected to autonomously monitor and control several aspects of our daily lives in almost all industrial sectors such as transportation, homes, environment, healthcare, to name a few. As a consequence of this pervasive monitoring, a huge volume of data is expected to be generated, which will ultimately demand for an efficient storage and processing technology, in such a way that the data could be remotely accessed and analyzed to support a smart decision-making process.

By abstracting the computing services as commodities, cloud computing emerged as a perfect match to fulfill that demand since it has been successfully dealing with data storage and processing at an incredible scale, in addition to enabling the application of analytics tools and information visualization techniques for data analysis. For instance, current cloud computing systems such as Microsoft Azure already comes with built-in tools for data analysis such as ML studio and Azure Data bricks [19]. Similar tools are presented by Amazon Web Services in its big data suite [20]. Nevertheless, in order to make this scenario

realistic, the IoT paradigm has to firstly accomplish a ubiquitous access through legacy systems, and especially, evolving wireless networks in such a way that the data transmission could be efficiently performed between the end devices and the cloud data centers. Toward this end, focusing on mobile applications and services, emerging 5G systems [21] have arisen. These systems will be featured by a massive deployment of small cells and macro cells coexisting with multiple cloud data centers which might belong to the MNO or a third party CSP. In such system, a close collaboration is expected to prevail between the MNOs in one side and the HetNets and the CSPs (called Intercloud) in the other side, with the goal to achieve the computational requirements and communication demands for mobile applications and services in an MCC environment.

This Chapter describes some background information on the integrated architecture (1.1) considered in this thesis. The main components of such architecture and their associated features from a resource management perspective are discussed. Next, the network simulator for 5G HetNet uplink that we have developed for evaluating the performance of the algorithms proposed in this thesis is described. Finally, a literature review of the most representative resource management schemes that have been proposed to boost the storage and computational capabilities of mobile devices by using the MCC, MEC and HetNet technologies is presented.

2.1 Background

2.1.1 The Need for Heterogeneous Wireless Networks

In terms of big data transmission, two emerging technologies are of paramount interest: machine-to-machine (M2M) communications and Mobile Cloud Computing (MCC). M2M communications is a promising technology for the development of IoT since it allows a seamless communication between machines without the need of human intervention. In turn, MCC is a merge between cloud computing

and mobile computing, which makes it viable for a device equipped with communication capabilities to offload its computing task to be processed and storage in the cloud.

The realization of M2M communications and MCC over wireless networks, particularly cellular mobile networks or 3GPP based systems such as LTE/LTE-A networks, is mostly conceived by means of scheduling algorithms [22], which are an integral part of the radio resource management procedures. These algorithms are responsible for allowing the M2M and Human-to-Human (H2H) traffic to share the wireless resources. In this regard, despite the progress made so far, there are still many open issues when it comes to designing a scheduling algorithm for a massive M2M deployment over cellular mobile networks. For instance, questions such as how to integrate M2M traffic and H2H services over the network and how to ensure per flow quality of service (QoS) naturally arise since scheduling algorithms in LTE/LTE-A networks [22], [23] are mostly based on random access channels that become congested when the number of devices requesting access in the network grows, as is the case of big data traffic. In this scenario, unless establishing strict policies to preserve the H2H service performance, the QoS of these applications may be degraded.

Furthermore, IoT traffic, which was prevalently non-real time in the past, is presently becoming delay tolerant since applications such as video surveillance and accident prevention are now prevalent. Hence, the radio resources have to be organized and allocated to deliver the designed QoS requirements regardless of the application service class (i.e. whether it is M2M or H2H service). Such constraints raise significant challenges in the system design because they demand for a real time access to M2M traffic. This has motivated the need for HetNets.

Considering the current literature focused on realizing M2M communication over wireless networks [24], it has been established that M2M traffic is carried out over a flat cellular structure, neglecting the fact that most current cellular mobile networks have heterogeneous, overlapping, and collaborative architectures. This

overlapping topology, often found in urban centers, stands by the name of HetNets.

In a HetNet, a set of RATs provides complementary services based on the inherent characteristic of each RAT. For instance, while small cells are able to support energy efficient data transmission with high rates at affordable prices, macro cells and micro cells have a mobility oriented design, where connections are seamlessly transferred between access points. Here, in order to be holistically operational, each RAT in a HetNet has its own radio resources included in a common pool of wireless channels, and the resource allocation of such channels is realized, aiming at the maximization of the overall system performance.

To achieve such goal, the 3rd Generation Partnership Project (3GPP) [6] introduced a procedure (called joint radio resource management (JRRM)) that collects the view of all available radio resources in the HetNet to enhance the overall QoS provisioning and resource utilization. As an example of how JRRM could be utilized, Saker et al. [25] proposed an eco-friendly HetNet design in which the cooperation among the available multiple RATs is exploited by a JRRM algorithm, which itself utilizes a switch on/off procedure on the base transceiver station (BTS) resources to impose which RAT has to be activated.

An example of HetNet is illustrated in Fig.2.1, which consists of a layer containing small cells (femtocells and picocells) on top of a microcell and/or microcell.

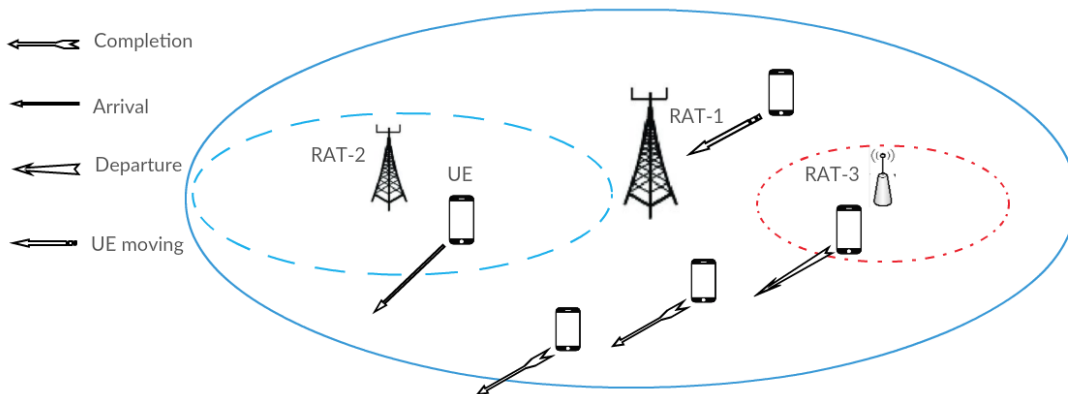


Figure 2.1: HetNet with three RATs and the associated random events

In Fig.2.1, the exploitation of the collaborative access in HetNet is a promising alternative to alleviate the burden of dealing with the ever-growing M2M traffic over cellular mobile networks. In this regard, scheduling algorithms can be designed and optimally utilized within the framework of the Common Radio Resource Management (CRRM) introduced by the 3GPP Consortium [6], in complement of using the mobile edge computing paradigm to help alleviating the latency problem encountered in the big data transmission from the UEs to the cloud data centers in a cloud-centric system.

2.1.2 MCC and Mobile Edge Computing in a Cloud-Centric System

The design of MCC solutions has also been facing the challenge of addressing the latency problem over a standalone wireless network in a flat architecture [26], which in the case of a real-time application for a computing system that requires real-time actions, may become a bottleneck. Considering a site with a large number of devices offloading their processing and storage computing tasks, not only latency will be a problem, but also the resource sharing with the H2H services. In this regard, it is believed that HetNet is a suitable enabler for massive data transmission over MCC in a cloud-centric system. For instance, a device may take advantage of the parallel transmissions offered by HetNet [5] to speed up the offloading process and reduce the latency over wireless networks as well as the impact of M2M communications over H2H services. Additionally, a real-time big data application may take advantage of a MEC server or Mobile Edge Computing (MEC) at the edge network level to speed up the computational tasks processing, thereby enhance the decision making process.

Recently, the issue of application management at the edge network for MEC in 5G systems [27] has attracted a lot of attention in the literature. In this regard, there has been a clear demand for latency sensitive applications with timely storage and computing resources. On the other hand, the effectiveness of MEC is

the cornerstone of the success of MCC applications [5], which in turn is tight up with the strategy to distribute the allocation of computing and communication resources at the edge network level [28]. Thus, resource management can also be considered one of the key aspects in the design of a MEC solution.

To address this problem from an algorithmic point of view, the joint communication and computation resource management schemes proposed in [4] are examples of solutions that can be considered. However, the velocity and variety of the big data traffic may still a threat to the service provisioning so that guaranteeing the strict latency control constraint of storage and computational hungry MCC applications becomes an issue that the MNO will have to deal with. Nevertheless, using MEC is still the best pathway to provide latency sensitive applications with timely storage and computing resources [5].

An example scenario of a MEC solution over 5G systems is illustrated in Fig.2.2, where the MEC server is designed to enable a one hop away connectivity

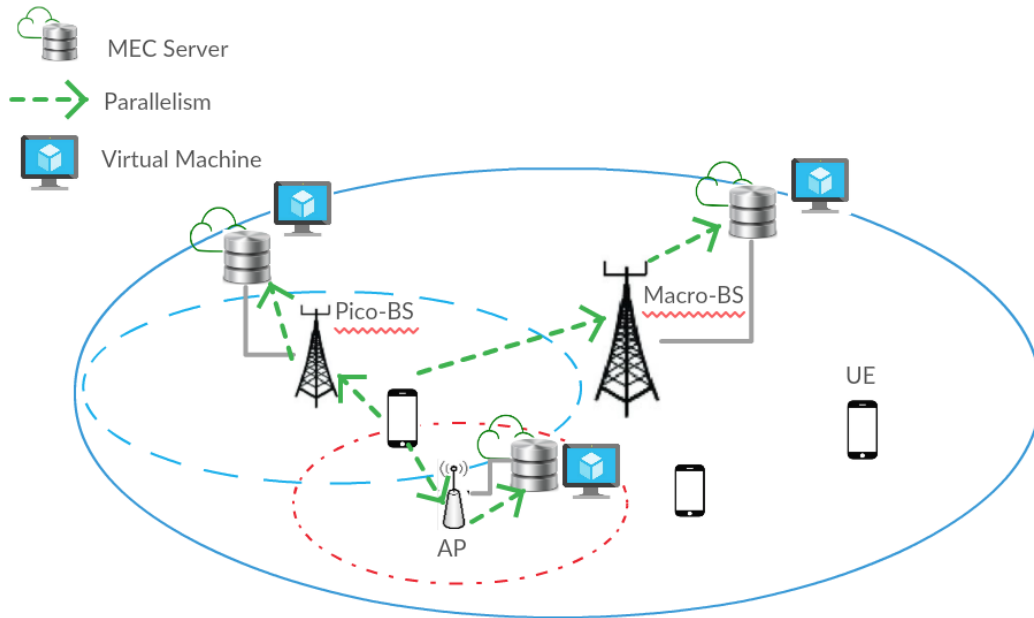


Figure 2.2: MEC over 5G systems

between the mobile users (UEs) and the edge-servers via a BS or an AP. Here, the UE segments the application into two parts and sends them to the two concurrent edge servers for processing, thereby speeding up the data transmission tasks. In doing such processing tasks, the MEC server might also benefit from the small-cell densification and the overlapping nature of the HetNet deployment, which enables the mobile users to be associated with the closest possible AP or BS.

2.1.3 Intercloud over HetNet for MCC Applications in Emerging 5G Systems: A Resource Management Perspective

The mobile users (UEs side), the Common Radio Resource Management Server (CRRMS) in the HetNet and the Cloud Service Broker (CSB) in the cloud side, constitute the main building blocks of the Intercloud over HetNet for MCC Applications in Emerging 5G Systems architecture (as illustrated in Fig.1.1). From a resource management standpoint, the interoperation between the Intercloud and HetNet requires that a collaboration in the form of several service level agreements prevails between the MNOs, CSPs, and UEs for resource sharing, tasks offloading and (VMs) allocation purposes [7].

In this integrated ecosystem, a common pool of wireless resources from HetNet and VMs from the cloud side will be made available for sharing. While the optimization of these wireless resources is operated by the CRRMS module via its intrinsic operations, namely, network selection, inter-system handoff and admission control, the optimal allocation/de-allocation of VM resources across multiple CSPs - to satisfy the UEs requirements in terms of heavy tasks processing - and the deployment and management of the application's execution over the successful provisioned VMs, are operated by the CSB (or group of CSBs) that will best satisfy the UEs requests.

It should also be noted that small tasks processing are typically handled

by the MEC servers at the edge network level; and a successful transmission of big data traffic from the UEs to cloud data centers is mainly attributed to the promotion of cooperation among the RATs, the exploitation of the intrinsic diversity in cell sizes and layers of HetNet, and the coordination of the UEs access to macro cells and large number of small cells, as orchestrated by the CRRMS [7]. Toward this end, an algorithmic framework defining how HetNets may be used jointly with cloud computing systems was proposed in [24], along with three possible approaches for achieving M2M communication over LTE/LTE-A systems [22].

In the first approach, a UE device may establish a direct link with the base station-eNodeB (eNB). In the second approach, M2M devices are organized into groups, each equipped with a M2M gateway which promotes the communication with the eNB in a multi-hop manner. In this case, a scheduling algorithm should be invoked to control the communications of M2M gateways under the coverage area rather. Since this solution is more manageable, group-based M2M communications is preferred as solution in case of a massive M2M deployment. In the third approach, the devices may communicate with each other using a peer-to-peer communication technique. In such case, a device may rely on other devices to enlarge its communication area.

The design aspects and challenges, as well as the development mismatch associated with the implementation details of Inter-Cloud architectures and application brokering, are described in some details in [7], [29].

2.1.4 Network Simulator for 5G HetNet up-link

The performance of all proposed methods have been validated through implementation on our developed simulated 5G up-link network model integrating the HetNet, MCC and MEC technologies. The network model is executed on Matlab 2018. The components of the network model are shown in Fig.2.3.

The details of these components are as follows:

2.1.4.1 System Parameters for 5G Up-Link

NR and LTE are integral parts of the 5G radio access technologies. LTE is designed to operate below 6 GHz frequencies while NR is envisioned to operate in the frequency ranging from 1 GHz up to 100 GHz [30]. The values of the system's parameters used in the simulator model are listed in Table 2.1.

Table 2.1: System parameters.

Abbreviation	Parameter description	Value.
P_m	Transmit power of macrocell BS	45 dBm
P_f	Transmit power of femtocell BS	20 dBm
P_{UE}	Transmit power of UE	23 dBm
η	Noise experienced by a UE (% of total interference experienced by UE)	10%
N_{PRB}	Number of PRBs available	100
B_{ch}	Channel bandwidth	20 MHz
Υ	Bandwidth per PRB	180 KHz
η_m	macro BS back-haul connection speed	500 Mbps
η_f	femto BS back-haul connection speed	100 Mbps

2.1.4.2 Location of Network Elements

As depicted in Fig.2.4, the network consists of two macro BSs with coverage areas denoted by blue circles, three femto BSs with coverage areas denoted by blue circles and a number of UEs denoted by red rhombuses.

Each BS has access to the public cloud system through a back-haul connection, and there exists a MEC server associated with each BS. The network serves a number of mobile devices randomly dispersed over the geographical area.



Figure 2.3: Overview of simulator

We have considered a set of N_{UE} UEs, each of which has a delay sensitive task of storing a big data size to the cloud storage system. Being a part of a dense HetNet, it is assumed that most of the UEs have access to multiple BSs. Details on the network elements used for our simulations are provided in Table 2.2.

Table 2.2: Location of network components

Abbreviation	Parameter description	Value.
N_{UE}^{in}	Number of indoor users	0
N_{UE}^{out}	Number of outdoor users	20
N_{UE}^{total}	Number of total users	20
N_m	Number of macrocells	2
N_{FAPs}	Number of femtocells	3
N_{iter}	Number of iterations	1000

For the sake of simplicity and traceability analysis, a quasi-static scenario is assumed, where the location of the UEs are considered to be unchanged during a single storage execution. In order to nullify the influence of the random locations of UEs on the network performance, a Monte-Carlo simulation method is adopted. Also, the simulation is performed for N_{iter} number of iterations and the values of

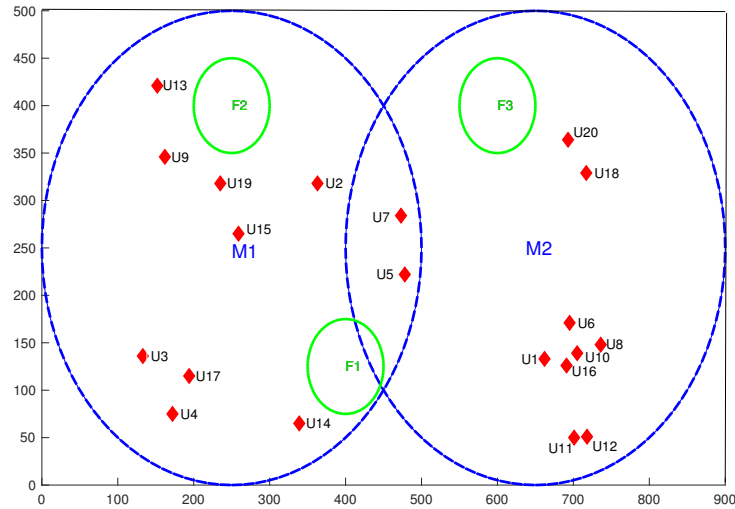


Figure 2.4: Location of network elements

the performance parameters calculated during these iterations are averaged.

2.1.4.3 Cell Association for all UEs: Biased Cell Association Method

Conventionally, in cellular networks, when a UE has to select a BS for transmission, it chooses the one from which it receives the maximum signal power. However, if the same strategy is applied to the HetNet architecture with both macro and small cells, this could lead to an under-utilization of the small cell BSs. Since the small cell BSs transmit at lower power compared to the macro BS, unless a UE is very close to the small cell BS, the signal strength from the macro BS is likely to be larger than that from small cell BS and most of the UEs will associate themselves to the macro BSs.

In order to increase the usability of small cell BSs, a cell range expansion (CRE) technique implementing the cell selection bias as proposed in LTE standards [31] is followed. In the proposed methods, since a UE associates itself to more than one BS, we have considered SINR) experienced by a UE instead of the received signal strength for the BS association decision. Therefore, for each UE, the BS association is based on the SINR experienced by the UE on the down-link transmission link. A UE associates itself to all BSs from which it experience a SINR higher than a prescribed SINR-threshold value (γ_{femto} or γ_{macro}). This SINR threshold value is assumed to be set by the network designer.

With the implementation of CRE, an additional term named as cell selection bias (α) is considered. Then, the SINR threshold for the association decision with the femto BS γ_{femto} is given by $\gamma_{femto} = \gamma_{macro} - \alpha$, where γ_{macro} is the SINR threshold for the BS association decision with the macro BS. Therefore at a time instant, a UE is associated with multiple BSs who can provide a SINR performance higher than the respective $\gamma_{femto}/\gamma_{macro}$ value. The values of the system parameters related to the cell association are listed in Table.2.3

Since a UE is likely to be associated with multiple BSs in the HetNet

Abbreviation	Parameter description	Value
γ_{th}^{Macro}	SINR threshold for cell association for macro BS	31 dBm
γ_{th}^{Femto}	SINR threshold for cell association for femto BS	-10 dBm
α	Cell association bias	15dBm

Table 2.3: System parameters related to cell association.

environment, it is likely that multiple paths will be available between the UE and the cloud storage system. For instance, the UE-1 from Fig.1.1 has the following data paths available to reach the cloud system:

1. UE-1 \rightarrow Macro BS-1 \rightarrow Macro BS-1 MEC server
2. UE-1 \rightarrow Macro BS-1 \rightarrow Back-haul Internet \rightarrow public cloud
3. UE-1 \rightarrow Macro BS-2 \rightarrow Macro BS-2 MEC server
4. UE-1 \rightarrow Macro BS-2 \rightarrow Back-haul Internet \rightarrow public cloud
5. UE-1 \rightarrow Femto BS-2 \rightarrow Back-haul Internet \rightarrow public cloud
6. UE-1 \rightarrow Femto BS-2 \rightarrow Femto BS-2 MEC server

2.1.4.4 Channel Model

In order to simulate the power and interference received by the UE and the macro/femto BS, we have considered the path loss model proposed in [32]. For the down-link transmission simulation, the power received by a UE is calculated as per Equations (2.1)-(2.2). The down-link simulation is used to determine cell association of UEs. For the up-link transmission simulation, the power received by a BS is calculated as per Equations (2.3)-(2.4). For the performance analysis of the proposed methods, the up-link transmission simulation is considered. The path loss (ϵ) is calculated by using Equations (2.5)-(2.8), depending on the location of the UE and the type of BS.

$$\hat{P} = P_m - \epsilon \quad (\text{dB}) \quad (\text{for macro base station}), \quad (2.1)$$

$$\hat{P} = P_f - \epsilon \quad (\text{dB}) \quad (\text{for femto base station}), \quad (2.2)$$

$$\hat{P} = P_{UE} - \epsilon \quad (\text{dB}) \quad (\text{for macro base station}), \quad (2.3)$$

$$\hat{P} = P_{UE} - \epsilon \quad (\text{dB}) \quad (\text{for femto base station}), \quad (2.4)$$

Besides, the path loss ϵ_f^{in} (in dB) between the indoor users and the femtocell BS is obtained as

$$\epsilon_f^{in} = 37 + \alpha_{in} \cdot 10 \cdot \log_{10}(d) + 18.3. \quad (2.5)$$

The path loss ϵ_m^{in} between the indoor users and the macrocell BS is obtained as

$$\epsilon_m^{in} = 40 + \alpha_{out} \cdot 10 \cdot \log_{10}(d) + pl_{bp}. \quad (2.6)$$

The path loss ϵ_m^{out} between the outdoor users and macrocell BS is obtained as

$$\epsilon_m^{out} = 40 + \alpha_{out} \cdot 10 \cdot \log_{10}(d). \quad (2.7)$$

The path loss ϵ_f^{out} between outdoor users and femto BS is obtained as

$$\epsilon_f^{out} = 37 + \alpha_{out} \cdot 10 \cdot \log_{10}(d) + pl_{bp}, \quad (2.8)$$

where d is the distance between the UE and the BS and the other network parameters are given in Table 2.4, P_f is the transmit power of the femto BS, P_m is the transmit power of the macro BS, P_{UE} is the transmit power of the UE, \hat{P} is the received power by a UE in case of down-link transmission and \hat{P} is the received power by a BS in case of up-link transmission. The path loss exponent (α) varies in the range [2-4], where 2 is for the propagation in free space and 4 for

the propagation in the noisy environment [33].

Table 2.4: System parameters related to channel model.

Abbreviation	Parameter description	Value
α_{out}	Outdoor path loss exponent	3.5
α_{in}	Indoor path loss exponent	4.5
pl_{bp}	Building penetration loss	8

2.1.4.5 PRB Allocation to Up-Link Data Paths

The LTE physical layer downlink transmission implements the Orthogonal Frequency Division Multiple Access (OFDMA) while the up-link transmission implements the Single Carrier Frequency Division Multiple Access (SC-FDMA). The OFDMA technology allows the data to be directed to/from multiple UEs on a sub-carrier basis for a specified number of symbol periods. In the LTE specifications [34], these basic sub-carrier and symbol period are referred to as a physical resource block (PRB), where a PRB is the smallest element of the resource allocation assigned to a data transmission link by the RAN.

As per the LTE up-link specifications [34], the number of PRBs available for the UE's data transmission with the total transmission bandwidth of 20 MHz is 100. These PRBs need to be allocated to all active data links of the network judiciously in order to maximize the network performance.

2.1.4.6 Calculation of the SINR, Data Rate and Other Desired Network Performance

The SINR (i.e. $\gamma_{i,j}$) between i^{th} UE and the j^{th} is calculated as per Equation (2.9). The maximum possible data rate on a wireless link is calculated as per Equation (2.10) following the Shannon-Hartley theorem [35]. The total interference experienced by the wireless link ($\tau_{i,j}$) is the sum of the received power from all BSs who shares at least one PRB allocated to the wireless link between the i^{th}

UE and the j^{th} BS as per Equation (2.11). The amount of bandwidth allocated to the wireless link ($W_{i,j}$) is the number of PRBs allocated to it ($\zeta_{i,j}$) multiplied by the bandwidth per PRB (Υ).

$$\gamma_{i,j} = \frac{\hat{P}_i^J}{\tau_{i,j} \cdot [1 + (\eta/100)]} \quad (2.9)$$

$$\beta_{i,j} = W_{i,j} \cdot \log_2(\gamma_{i,j}) \quad (2.10)$$

$$\tau_{i,j} = \left(\sum_{k=1, k \in \bigcup_{i=1}^{N_{UE}^{total}} \{\zeta_i\}, k \neq j}^{N_{BS}^{total}} \hat{P}_i^j \right) \quad (2.11)$$

$$W_{i,j} = |\zeta_{i,j}| \cdot \Upsilon \quad (2.12)$$

2.2 Literature Review

Due to the explosive usage of mobile devices such as smart phones, wearable devices, smart cars, tablets, to name a few, there is a huge increase in demands for computational and data storage capabilities of UEs. MCC and HetNet technologies have been used as a solution to amplify these high data storage and computational capabilities of mobile devices while following constraints of low device size, low device weight and low energy consumption.

The resource management and task scheduling of the network embedding these technologies is critical in order to exploit the full potential of these technologies to achieve the desired network performance. In the literature, various solutions have been proposed with an attempt to boost the storage and computational capabilities of mobile devices by use of these technologies. The scheduling and management of the task offloading process to cloud centers and the management of the limited radio resources is critical for achieving the performance maximization of MCC networks.

From a resource management perspective, the intrinsic diversity of technolo-

gies in wireless and cloud domains arises as an attractive feature to be exploited in order to attain the desired network performance. The problem of resource management has been extensively researched in the field of MCC networks.

In [4], Barbarossa et al. considered a network consisting of distributed cloud computing architecture over a HetNet, where under the normal working conditions, a service request is assigned to the closest cloud in order to minimize both the latency and energy consumption. However, in case the nearest cloud is unable to meet the service request, the second nearest cloud is solicited, and so on, until the service request is served. Under this architecture, a radio and computational resource allocation technique is devised, which manages the resources at the signal processing level, with the goal to reduce the energy consumption and latency.

In [7], Gkatzikis et al. investigated the role that the Virtual Machine (VM) migration plays in an architecture composed of multiple CSPs enhanced with a MEC infrastructure, aiming at improving the service provisioning. They proposed three VM migration strategies: load-aware strategy, no-migration strategy, and load-and mobility-aware strategy. Simulation results showed that the load-and mobility-aware strategy yields the shortest task lifetime compared to the other strategies. In addition, the benefits of each type of task decision making (i.e. task decision initiated by the server, task decision task decision initiated by the CSP, or task decision from the task itself) are compared, showing that the CSP initiated task offloading strategy achieves the best performance. On the other hand, Felemban et al. [36] proposed an integrated MCC-MEC architecture for multimedia services, which can be utilized to transfer the multimedia information from the multimedia data centers' storage to the UEs via MEC virtualized servers located at the edge network, with shortened latency.

In [8], Carvalho et al. provided an analysis of the interoperation between HetNets and Intercloud, leading to a description of the main components of an Intercloud over Hetnet architecture from a resource management standpoint. Related challenges and performance criteria that must be taken into account in the

system design are discussed, along with the nature of operations. An approach to optimize the revenue sharing among the MNOs and CSPs when cooperatively attending a given service region is also proposed, which relies on the concept of Shapley value, considering the radio resources from the HetNet and the VM resources from the Intercloud. However, in the implemented resource allocation model, only applications that have similar requirements in terms of computing and communications are considered. A comprehensive summary of related works on MCC is provided in [5].

In [9], Chen et al. proposed a game theoretic-based solution for achieving efficient computational intensive task offloading for MCC networks. A decentralized computational task offloading problem among mobile device users was formulated as a decentralized task offloading game considering both computation and communication aspects. By analyzing the game structure, the authors proposed a decentralized computation offloading mechanism which can achieve a Nash equilibrium of the game. Their results demonstrated that the proposed solution is efficient and scalable.

In [10], Liang et al. proposed a infrastructure resource allocation scheme in a multi-cloud ecosystem whose objective is to maximize the reward for the CSP and minimize the service disruptions for the users. The proposed Semi Markov Decision Process (SMDP)-based solution decides on whether to accept or reject a service request in the home cloud or transfer it to an adjacent cloud, by jointly considering the system's incomes and expenses. Through simulations, the superiority of the proposed method compared to the Greedy policy is proven.

In [37], Kaewpuang et al. presented the performance analysis of a network featured by a coalition formed by multiple mobile CSPs sharing their radio and computing resources with an objective to maximize their revenues. The resource management and the revenue sharing problem is formulated as a linear programming, stochastic programming, and robust optimization problems, respectively. Using the concept of Shapley value, a revenue management approach is proposed,

which determines the revenue to be shared among the mobile CSPs. In addition, a game theoretical scheme is also proposed to determine whether a CSP should participate in the coalition or not. Simulation results are provided to strengthen the benefits of multiple mobile CSPs coalition in terms of increased respective revenues.

In [38], Choi et al. investigated the performance of the MCC network through intelligent scheduling and management of the task offloading process. They exploited the parallel uplink transmissions in a HetNet over multiple RATs to address the issues of air interface selection and band selection for power allocation to the chosen links. Their proposed distributed joint allocation scheme is shown to maximize the total system capacity. Through simulations, a performance comparison with a single RAT transmission method yields an improved capacity achieved by parallel transmission in HetNet.

In [39], Kosta et al. proposed a framework for offloading the computational intensive tasks to the cloud centers in the MCC network. The capabilities of UE are boosted by a method level computation offloading and parallelizing method execution using multiple VMs' images. The performance of the proposed framework was evaluated over a range of benchmark applications and the simulation results have proved a significant improvement in terms of execution time and energy consumption.

In [40], Zhang et al. considered a dense deployment of HetNet as a solution to support the expected massive growth of mobile data in the upcoming 5G networks. Many challenges in terms of network operation and resource management, cost effective small cell deployment, and intercell interference mitigation, are identified; and a cloud based platform is introduced to tackle these issues. Finally, its benefits are illustrated by means of two case studies.

In [41], Shiraz et al. provided a critical analysis of the resource-intensive nature of existing computational offloading techniques for MCC networks. A prototype application is evaluated with varied computation intensities in a real

MCC environment. Simulations results show that additional computing resources are consumed by the deployment of the computational tasks offloading to cloud centers at runtime. Moreover, several aspects that should be taken into account during the design of efficient tasks offloading techniques in MCC networks are discussed.

In [42], Chen et al. focused on the reliability and energy efficiency during data storage and processing in mobile cloud data centers, and they proposed a scheme to efficiently achieve these performance goals under energy efficiency consideration. The proposed scheme splits the data into chunks and assigns these chunks to dedicated nodes in such a way that other nodes can reliably retrieve them with minimal energy consumption. The proposed scheme is validated on a real hardware setting, showing its effectiveness.

In [43], Gerasimenko et al. introduced the concept of heterogeneous cloud radio access network (H-CRAN) emerged as a cost effective solution to improve the available cooperative gains in HetNets. They addressed the problem of cooperative radio resource management in 5G-grade H-CRANs and proposed a comprehensive resource optimization and control method for real time performance optimization of these networks. Simulation results show that the proposed solution can achieved a balance between the system's throughput and fairness of the resulting resource allocations.

In [44], Liu proposed a technique that enables the UE to partition a computational intensive task into sub-tasks before uploading them into multiple MEC servers. An optimization problem is formulated to jointly minimize the latency and offloading failure probability in order to achieve an ultra reliable low latency communication. Since the formulated problem is non-convex, the proposed solution algorithms is based on heuristic search, reformulation linearization technique and semi-definite relaxation. Through simulations, the proposed solutions are shown to achieve the desired balance between the latency and reliability.

In [45], Zhao et al. focused on the minimization of energy consumption on

the UEs and proposed a scheme to jointly optimize the radio resource and computational resource allocation. The energy consumption minimization problem is also formulated as a mixed integer nonlinear programming problem, which is subject to latency constraints specific to targeted applications. Two algorithmic solutions are proposed to solve the problem, yielding a significant reduction in the UEs' energy consumption.

In [46], Zhang et al. focuses on minimizing the energy consumed during the task offloading in 5G HetNets equipped with MEC servers. A task offloading scheme is proposed, which jointly optimizes the task offloading and radio resource allocation to minimize the energy consumption under the latency constraints. Through simulations, the proposed schemes are shown to be energy efficient.

All the above work have focused on the performance enhancement of the MCC/MEC network implementing the offloading of computational intensive tasks. These work have considered only the computational intensive task offloading as main objective. To the best of our knowledge, the issues related to the process of storage intensive task offloading has not yet been considered so far. Our system model incorporates both centralized cloud data centers and MEC. Moreover, our proposed methods are designed in such a way as to exploit the features of the coverage from multiple BSs that are available in the dense deployment of HetNet, with focus on reducing the data transmission delay while transmitting big data files to the cloud data storage through an efficient utilization of the enhanced infrastructure of 5G HetNet and intelligent allocation of limited radio resources.

The method proposed in Chapter 3 splits the data file to be transmitted into a number of data chunks, where the number of data chunks is equal to the number of data paths available to the UE under consideration. These data file chunks along with the associated lost data recovery syndromes are transmitted on the available data paths simultaneously.

The algorithm described in Chapter 4 considers multiple UEs active in the network and focuses on the performance improvement of the overall network users

through the management of the radio and infrastructure resources. A game theoretical approach is devised in order to solve the problems related to resource management and task offloading scheduling process [47, 9]. We follow a mechanism design approach, a sub-field of game theory where the algorithm designer determines the rules of the game. The game theoretical approaches proposed in the literature formulated the game and proposed algorithms to achieve a Nash equilibrium. None of these approaches considered the factor of individual rationality i.e. the motivation for the players to continue to participate in the game rather than quitting it. In our proposed method, in addition to infrastructure and radio resource management algorithm, a pricing algorithm is proposed to ensure that the players/UEs continue to participate in the coalition.

The solution algorithm presented in Chapter 5 attempts to optimize the MCC network overlaid over 5G HetNet network, considering two important performance parameters, which are the data transmission latency and the energy consumption. The problem is formulated as a combinatorial optimization problem and an ACO based algorithm is proposed as solution to the problem.

Chapter 3

Efficient Methods for Mobile Big Data Transfer Over HetNet in Emerging 5G Systems

3.1 Introduction

One of the most important challenges faced by mobile devices while satisfying the high performance demands in current LTE and upcoming 5G networks is the limited resource in terms of device battery life, storage space, transmission bandwidth, to name a few. As stated earlier in Chapter 2, to meet the ever-increasing demand for these resources, MNOs have resorted to three complementary solution technologies, namely: HetNet, MCC, and MEC.

First, a HetNet is usually deployed to increase the overall network coverage as well as the spectral efficiency through higher frequency reuse. Second, MCC, which integrates cloud computing into a wireless network, is a promising technology to augment the UE capabilities in emerging 5G systems by wirelessly transferring the computation burden from it to the resource-rich cloud computing centers. Third, MEC is a technology that further amplifies the performance of

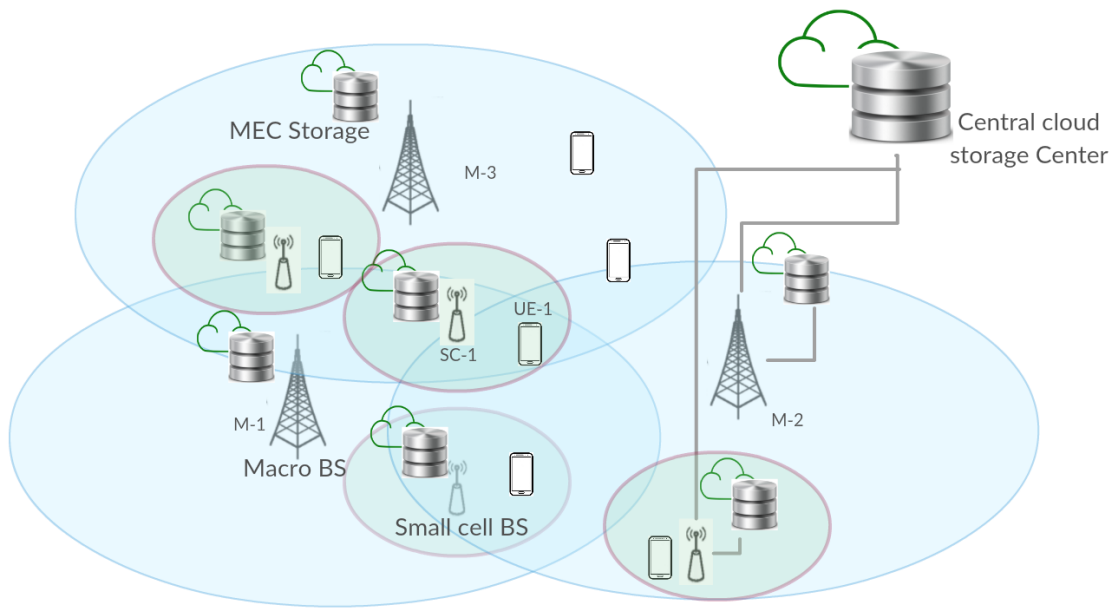


Figure 3.1: System Model.

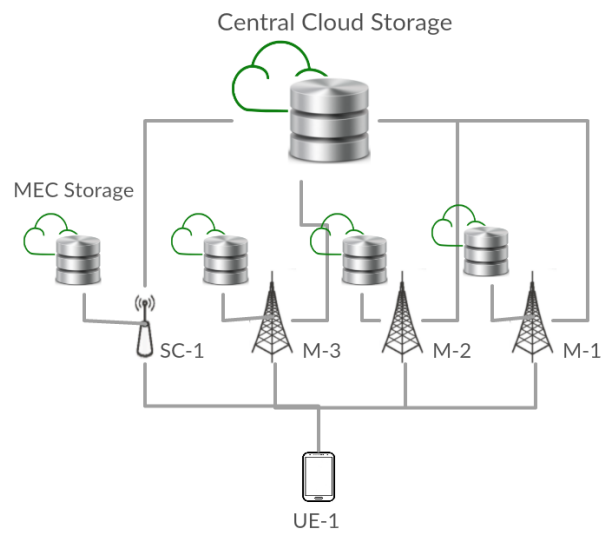


Figure 3.2: Data paths available between UE-1 and the cloud system in fig. 3.1.

MCC by implementing small scale cloud data centers (i.e. MEC servers) at the radio access technologies/networks (RATs) facilities, as closer as possible to the UEs, in such a way as to provide a powerful computing and storage resources to mobile devices with low latency. In this setting, the massive data transfer over a standalone wireless network is a challenging task due to the expected increased latency over the wireless channels associated with the remote cloud data centers.

In this Chapter, we have considered a wireless network architecture depicted in Fig. 3.1, an instance of the generic system model introduced in Fig. 1.1, which consists of multiple macro BSs, femto BSs, back-haul connection to a public cloud system, MEC servers associated with each BS, and a number of UEs. In this architecture, most of the UEs have multiple data paths to the cloud storage centers. As an example, Fig. 3.2 shows an instance where UE-1 has eight different data paths to the cloud data centers.

First, we have proposed a novel RAID-based data storage algorithm [11] that minimizes the data transmission delay during transmission of a big data file from the UE to the cloud data center. This algorithm exploits the multiple data paths available between the mobile device's UE and the cloud system, and it is inspired by the Redundant Array of Inexpensive Disks (RAID) technique [12]. Further details on the network architecture, cell association of UEs, locations of all network elements, have been provided earlier in Subsection 2.1.4 of Chapter 2. In particular, the cell range expansion technique [18] (i.e. biased cell association method) is utilized to increase the usability of small cell BSs. In the proposed algorithm, a lost data recovery technique is also integrated in order to deal with the problem of failure of a data chunk retrieval during transmission. In this regard, it is worth mentioning that the RAID technique combines multiple physical disk drive components into a single logical unit for the purpose of data redundancy. Here, two methods for big data file chunking are considered, (1) the big data file is equally split into data chunks of equal size, each of which is transmitted on one data link (This is referred to as *Case of Equal Data Chunk Size*); and (2) the

big data file is split into data chunks, where each data chunk size is proportional to the uplink capacity of the data link on which it is to be transmitted (This is referred to as Case of Data Chunk Size Proportional to uplink Capacity) [13].

Second, we have applied the proposed algorithm to achieve an efficient transfer of big video data files [48], where the H.265 compression algorithm is invoked to compress the video file before transmitting it over the network from the UE to the cloud data center.

Third, using our developed network simulator for 5G HetNet uplink, we have evaluated the effectiveness of the proposed algorithms in comparison to that of the conventional method, in terms of transfer delay and time required by a UE to store the file. By conventional method, we mean that a big data file of similar size is transferred to the cloud data storage via the macro BS with the highest SINR value using the back-haul connection.

3.2 RAID Based Algorithm for Mobile Big Data Transfer Over HetNet in 5G Systems: Case of Equal Data Chunk Size

Following the operational principle of the RAID system [12], our intend is to exploit multiple BSs to increase the data transfer rate from the UE to the cloud data storage and vice versa. As per the LTE-A standards [34], the network is designed by considering the system parameters provided in Table 3.1.

It is assumed that each UE has a big data file of a particular size and it executes Algorithm 1 on it. The number of data paths available to i^{th} UE are calculated as per equation 3.1, where N_i^{UEs} are number of BSs to which i^{th} UE is associated to. A big data file to be transmitted is split into N_i^{chunks} number of chunks as 2 data paths are reserved for the transmission of lost data recovery syndromes.

Table 3.1: System parameters.

Parameter	D escription	Value.
Ptx_m	Transmit power of macrocell BS	45 dBm
Ptx_f	Transmit power of femtocell BS	20 dBm
η	Noise experienced by a UE (% of total interference experienced by UE)	10%
N_{PRB}	Number of PRBs available	100
B_{ch}	Channel bandwidth	20 MHz
N_{SC}^{total}	Bandwidth per PRB	180 KHz
N_{UE}^{in}	Number of indoor users	0
N_{UE}^{out}	Number of outdoor users	20
N_{UE}^{total}	Number of total users	20
N_m	Number of macrocells	2
N_{FAPs}	Number of femtocells	3
d_f	Indoor penetration factor	1
α_{out}	Outdoor path loss exponent	3.5
α_{in}	Indoor path loss exponent	2.5
pl_{bp}	Building penetration loss	8
γ_{th}^{macro}	SINR threshold for cell association for macro BS	28 dBm
γ_{th}^{femto}	SINR threshold for cell association for femto BS	-13 dBm
\mathcal{Q}_m	Macro BS back-haul connection speed	500 Mbps
\mathcal{Q}_f	Femto BS back-haul connection speed	100 Mbps

$$N_i^{paths} = 2 \cdot N_i^{BSs} \quad (3.1)$$

$$N_i^{chunks} = N_i^{paths} - 2 \quad (3.2)$$

The total data transmission delay experienced by a UE is the maximum of the delays experienced by the data chunks traversing via data paths available to i^{th} UE as per equation 3.3.

$$DT_i^{total} = \max_{k=1}^{N_i^{paths}} DT_{k,i}^{total} \quad (3.3)$$

Algorithm 1 RAID based algorithm for faster data transfer: Case of Equal Data Chunk Size

- 1: Set the system parameters to suitable values as per Table 3.1 and set $N_{iter} = 100$.
 - 2: **for** iteration counter=0 to N_{iter} **do**
 - 3: Randomly place the UEs across the area.
 - 4: **for** UE index=0 to N_{UE}^{total} **do**
 - 5: Calculate the SINR experienced by each UE in downlink direction.
 - 6: Associate a UE to all macro BSs whose SINR is higher than macro SINR threshold.
 - 7: Associate a UE to all femto BSs whose SINR is higher than femto SINR threshold.
 - 8: Calculate the SINR experienced by each UE and its associated BSs in uplink direction.
 - 9: Divide a data file into N_i^{chunks} (from 3.2) number of chunks of equal sizes.
 - 10: Calculate the data transfer delay as per Equation (3.4) for each data path between the UE and the cloud system.
 - 11: Calculate the data transfer delay incurred by the UE = max(delay incurred to transmit each data chunk).
 - 12: **end for**
 - 13: **end for**
 - 14: Calculate the average data transfer delay incurred by each UE.
 - 15: Calculate the average data transfer delay incurred by a UE over the number of Monte-Carlo simulation cycles.
-

$$DT_{i,j}^{total} = \begin{cases} DT_{i,j}^{wireless} & \text{- If the data path is between the UE and its associated BS's MEC server.} \\ DT_{i,j}^{wireless} + DT_i^{back-haul} & \text{- If the data path is between the UE and the public cloud data center.} \end{cases} \quad (3.4)$$

The total transmission delay experienced by the j^{th} data path associated with i^{th} UE ($DT_{i,j}^{total}$) is the delay incurred on wireless data path ($DT_{i,j}^{wireless}$) if the data path connects to a MEC server. Otherwise if the data path connects to a public cloud data center, then the total delay is sum of delays incurred on the wireless link between UE and corresponding BS ($DT_{i,j}^{wireless}$) and the one incurred on the

back-haul connection from BS to the cloud center ($DT_i^{back-haul}$).

$$DT_{i,j}^{wireless} = \frac{S}{N_i^{chunks} \cdot \beta_{i,j}} \quad (3.5)$$

$$DT_{i,j}^{back-haul} = \frac{S}{N_i^{chunks} \cdot \varrho_m} \quad \text{for the macro BS back-haul} \quad (3.6)$$

$$DT_{i,j}^{back-haul} = \frac{S}{N_i^{chunks} \cdot \varrho_f} \quad \text{for the femto BS back-haul} \quad (3.7)$$

$$(3.8)$$

These delays are calculated as per the equations (3.5- 3.7), where S is the size of a big data file to be transmitted, ϱ_m is the macro BS back-haul connection speed, ϱ_f is the femto BS back-haul connection speed and β_i is throughput experienced on the wireless link which is calculated as per equation (3.9).

$$\beta_{i,j} = W_{i,j} \cdot \log_2 (\gamma_{i,j}) \quad (3.9)$$

$$\gamma_{i,j} = \frac{\hat{P}_i^j}{\left(\sum_{j=1, j \neq J}^{N_{BS}^{total}} \hat{P}_i^j \right) \cdot [1 + (\eta / 100)]} \quad (3.10)$$

For the uplink channel simulation, the SINR experienced by the i^{th} UE (γ_i) on the j^{th} data path is calculated using Equation (3.10). where \hat{P}_i^j is the power received by the i^{th} UE (γ_i) from the j^{th} BS and η is the noise level perceived as a percent of the total interference experienced by the UE. The possible bit rate on the wireless link between the i^{th} UE and the j^{th} BS is obtained using Equation (3.9), where $W_{i,j}$ is the amount of bandwidth available to j^{th} data path associated with i^{th} UE which is described in section 3.2.1.

3.2.1 Calculation of the Bandwidth Available to Each UE

As stated earlier in Chapter 2, the number of PRBs available with the total transmission bandwidth of 20 MHz (as per the LTE uplink specifications [34])

is 100. Here, we assume a frequency reuse factor of 1. On the other hand, the intercell interference is managed by using a fractional power control as described in [49]. Consequently, each UE has 100 PRBs available for up-link data transmission. Each PRB consists of 12 consecutive sub-carriers for one time slot, i.e. 0.5 msec duration. With a sub-carrier spacing of 15 KHz, each PRB has a bandwidth of $12 \times 15 = 180$ KHz. In addition, each UE equally allocates the available PRBs to its associated data paths. Therefore, the number of PRBs available to a data path is $100 / N_i^{paths}$, where N_i^{paths} is the number of data paths to i^{th} UE.

3.2.2 Lost Data Recovery Technique

When a storage is distributed among multiple cloud storage centers, the probability that one of these devices fails increases. Moreover, the data chunks are transmitted on unreliable wireless data links, it is possible to get a higher rate of failure. In our proposed algorithm, if the retrieval of one of the chunks fails, the file would be considered as corrupted. To overcome this, data error correction methods can be used. In this thesis, we have introduced a data correction method inspired from the RAID-6 system [50], which provides the protection against the failure to retrieve the data chunks. Two syndromes are calculated as per Equations (3.11-3.12), where N_i^{UEs} is the number of BSs associated with the i^{th} UE, then saved to the cloud data center along with the data chunks of the file.

Following our network design, each BS has an associated cloudlet. Therefore, each BS has two data paths to the cloud data center, one to the cloudlet and the other through the back-haul to public cloud system. Indeed, the number of total paths available for a UE to reach the cloud storage system are twice the number of BSs associated with that particular UE. Out of the total available number of paths, two paths are reserved for the transmission of the P and Q syndromes, obtained via Equations (3.11) and (3.12). Therefore, the number of data links available for the actual data transfer is $N_i^{paths} - 2$.

$$\begin{aligned}
P_1 &= D_1 \quad \text{where, } D_i \text{ is } i^{th} \text{ data chunk.} \\
P_i &= P_{i-1} \oplus D_i, \quad \text{for } 2 \leq i \leq N_i^{chunks} \\
P &= P_i \quad \text{where } i = N_i^{chunks}
\end{aligned} \tag{3.11}$$

$$\begin{aligned}
Q_1 &= g^1 \cdot D_1 \\
Q_i &= Q_{i-1} \oplus g^i \cdot D_i \quad \text{where } i = N_i^{chunks} \\
Q &= Q_i \quad \text{where } i = N_i^{chunks}
\end{aligned} \tag{3.12}$$

where the Galois field $GF(2^8)$ with generator g is considered. In calculating the syndromes P and Q , we have implemented a linear feedback shift register as field generator. Next, a big data file is equally divided into same sized data chunks D_i , each of which is transmitted on one data link. Therefore, for a data file of size S , the size of D_i is S / N_i^{chunks} , where N_i^{chunks} is the number of data paths available between the i^{th} UE and the cloud data center. The N_i^{paths} is calculated as per Equation (3.1).

3.3 Simulation Results: Case of Equal Data Chunk Size

In this section, the simulation results showing the performance of the proposed RAID based algorithm in the case of equal data chunk size is presented. The simulated network environment is depicted in Fig. 3.3, where the location of the femto and BSs are predetermined. However, the location of the UEs are randomly generated in the considered area. In order to nullify the impact of the UEs' location on the performance of the proposed algorithm, we have followed the Monte Carlo simulation technique.

The proposed algorithm is executed for a certain number (N_{ite}) of iterations,

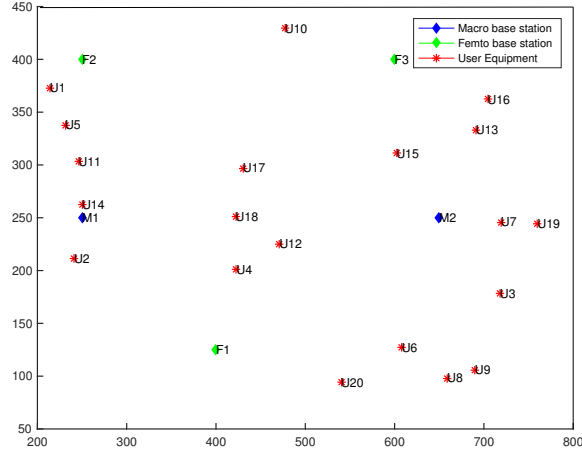


Figure 3.3: Simulated location of the HetNet elements.

under varying user locations for each iteration, and the performance measures obtained at each iteration are averaged. For comparison purpose, we have considered the conventional method, which consists of transmitting a big data file of same size to the cloud data storage via the macro BS with the highest SINR value using the back-haul connection, assuming that each UE in the network has a big data file of fixed size. The fundamental goal of our proposed algorithm is to increase the speed of the data transmission during the big data storage to the cloud data center.

First, the data file size is increased from 1 Gbits up to 10 Gbits and the impact of this variation on the the transfer delay, i.e. the time (in seconds) required to transfer the corresponding data file size while maintaining the same number of UEs and BSs is investigated. The results are captured in Fig. 3.4. As expected, as the size of the data file increases, the time required to transfer the file also increases. Also, the time required to transfer a file increases linearly as the file size increases when transmitted by the conventional method. However, when the same file is transmitted by our proposed algorithm, the time required to transmit the file is much lower. It is also observed that as the data file size

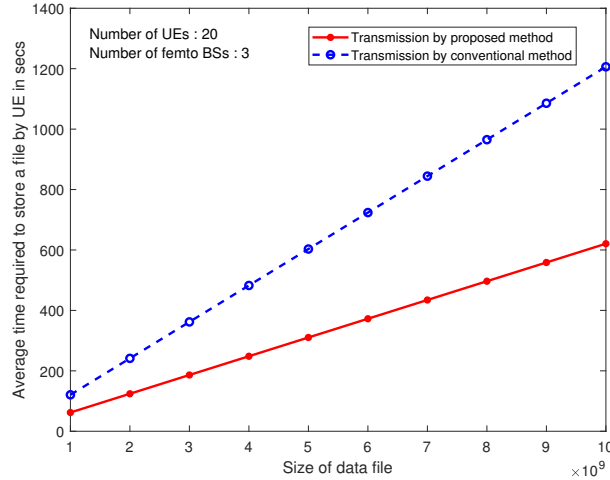


Figure 3.4: Data transfer delay vs. data file size.

increases, the time required to transfer it by our proposed method is significantly reduced. For instance, for a file of 10 Gbits size, the time required to transfer it using our proposed method is almost half the time required to send the same file using the conventional method.

Second, the number of UEs in the network is varied from 11 to 20 and the impact of this variation on the time (in seconds) required to transfer a data file of 1Gbits is investigated. The results are captured in Fig. 3.5. In Fig. 3.5, it can be observed that when transmitted by the conventional method, the time required to transfer the file increases linearly with the number of UEs. However, the performance of our proposed method is insensitive to the number of UEs, but slightly varies with the locations of the UEs.

Third, the number of femto BSs in the network is varied from 3 to 10 and the impact of this variation on the time (in seconds) required to transfer a data file of 1Gbits is investigated. The results are captured in Fig. 3.6. In Fig. 3.6, it can be observed that the performance of our proposed algorithm significantly improves with the increased number of femto BSs in the network. By increasing the number of femto BSs from 3 to 5, the time required to transfer the 1 Gbits data

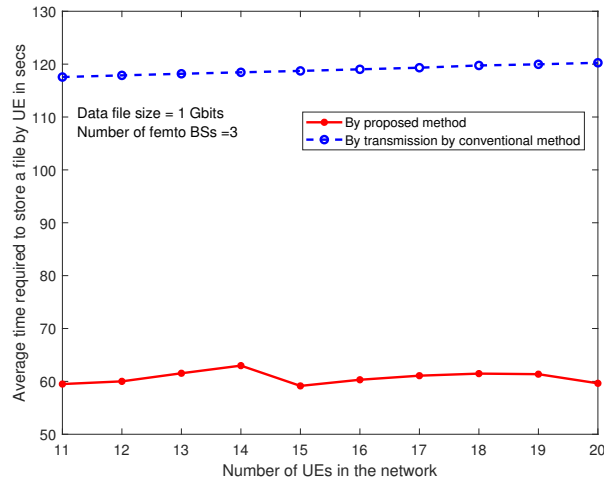


Figure 3.5: Data transfer delay vs number of UEs.

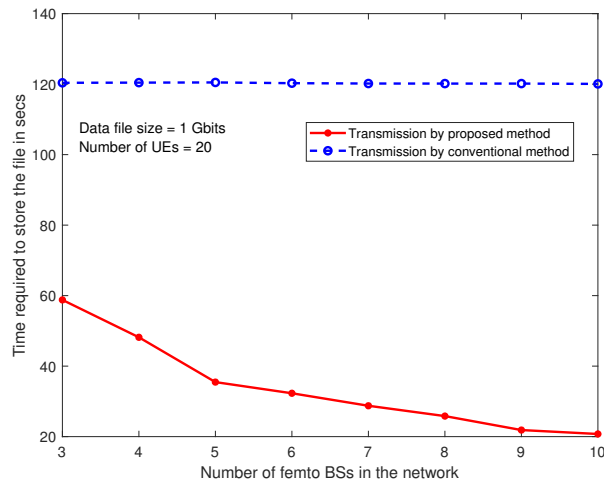


Figure 3.6: Data transfer delay vs. number of femto BSs.

file reduces by about 20 seconds. Furthermore, when the number of femto BSs is continuously increased, the time required to transfer the same file is continuously reduced.

3.4 RAID Based Algorithm for Mobile Big Data Transfer Over HetNet in 5G Systems: Case of Data Chunk Size Proportional to Uplink Capacity

In this proposed version of the RAID based algorithm, the size of a data chunk is directly proportional to the highest data rate that the wireless data link can support. This data chunk is to be transmitted on that data link.

For the considering network design model (shown in Fig. 3.1), each BS is equipped with a MEC server and it has two data paths to the cloud data storage, namely, one path going through the back-haul Internet connection to the public cloud system and the other path going to the MEC server. Thus, the number of paths N_i^{paths} that the i^{th} UE can use to reach the cloud storage system is twice the number N_i^{UEs} of BSs associated with that UE. As the first step of the proposed algorithm, the big data file to be transferred to cloud storage is split into N_i^{chunks} , where the size s_i^j of the i^{th} chunk to be transmitted on the data link passing through the j^{th} BS is calculated as:

$$N_i^{paths} = 2 \cdot N_i^{UEs} \quad (3.13)$$

$$N_i^{chunks} = N_i^{paths} - 2 \quad (3.14)$$

$$s_i^j = \frac{S \cdot (\beta_i^J - \beta_i^{min})}{\beta_i^{sum} \cdot (\beta_i^{max} - \beta_i^{min})} \quad (3.15)$$

The right hand side of Equation 3.2 is justified by the fact that in provision to transmit the syndromes associated with the data recovery technique that can protect against a failure to retrieve the data chunks in the cloud, two of the total available paths are reserved for the transmission of these syndromes. The proposed data transfer algorithm, whose input is a big data file of a certain size originated from a UE, is shown in Fig. 3.7.

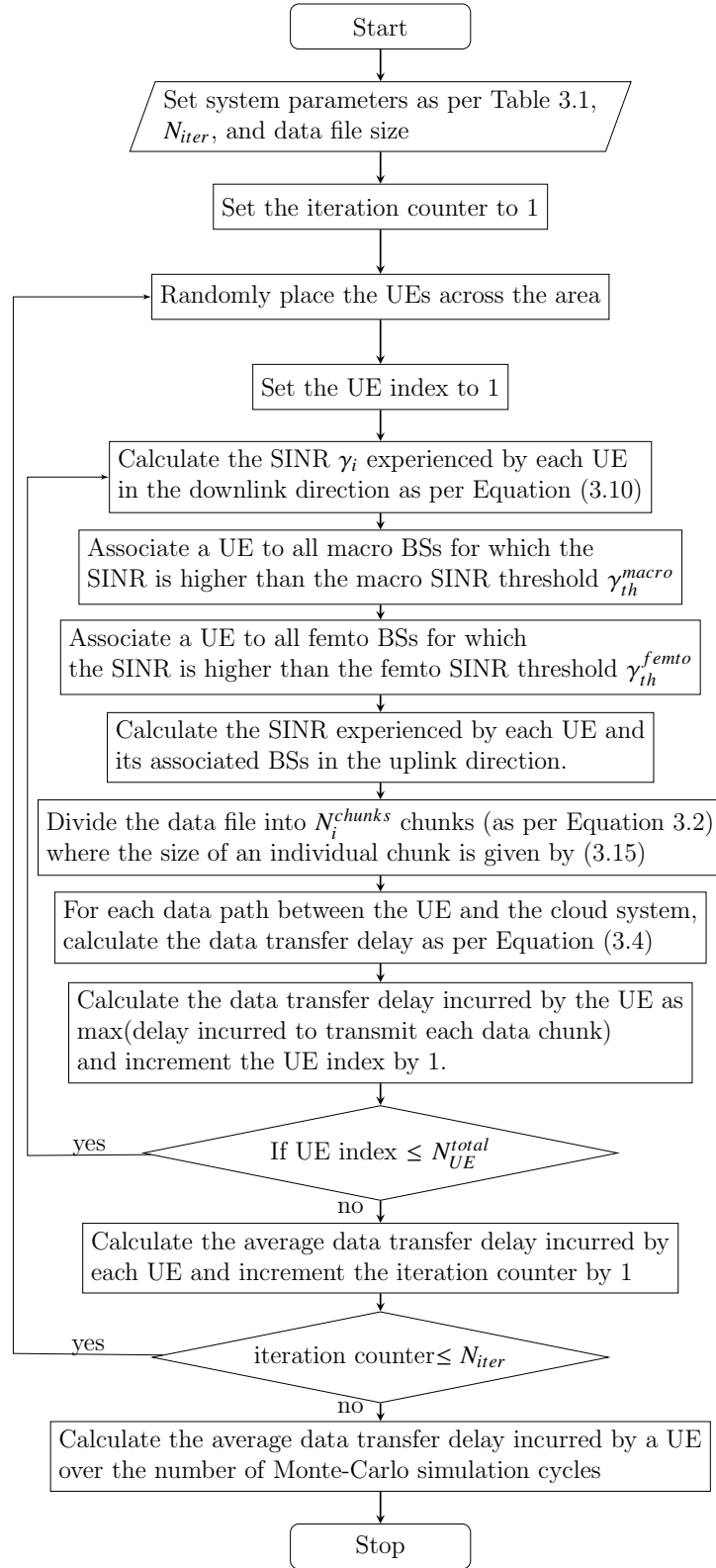


Figure 3.7: RAID-based algorithm for faster data transfer: Case of data chunk size proportional to uplink capacity

3.4.1 Lost Data Recovery Technique

The aforementioned lost data recovery technique is replicated, but this time, it is assumed that: (1) the size of the data chunk is proportional to the uplink capacity of the data link on which it is to be transmitted; and (2) the size of the considered syndrome P (resp. Q) is that of the largest data chunk (the smaller data chunks padded with zeros as per Equation (3.17)). These syndromes are calculated using Equations (3.18-3.19), then they are saved to the cloud system along with the data chunks of the file, i.e.

$$Z = \max \text{size}(D_i) \quad (3.16)$$

$$PD_i(k) = \begin{cases} D_i(k) & \text{for } k \leq \text{size}(D_i) \\ 0 & \text{for } \text{size}(D_i) < k \leq Z \end{cases} \quad (3.17)$$

$$P_1 = P \cdot D_1 \quad \text{where, } D_i \text{ is } i^{th} \text{ data chunk.}$$

$$P_i = P_{i-1} \oplus PD_i, \quad \text{for } 2 \leq i \leq N_i^{chunks}$$

$$P = P_i \quad \text{where } i = N_i^{chunks} \quad (3.18)$$

$$Q_1 = g^1 \cdot PD_1$$

$$Q_i = Q_{i-1} \oplus g^i \cdot PD_i \quad \text{where } i = N_i^{chunks}$$

$$Q = Q_i \quad \text{where } i = N_i^{chunks} \quad (3.19)$$

$$(3.20)$$

In Equation 3.19, it is assumed that in case a data chunk is loss, the UE to which it pertains is made aware of the size of this loss item. Therefore, the zero padded calculated chunk can be truncated in order to recover the original lost data chunk.

3.4.2 Simulation Results: Case of Data Chunk Size Proportional to Uplink Capacity

In this section, the simulation results showing the performance of the proposed RAID based algorithm in the case that the data chunk size is proportional to the uplink capacity is presented. The same simulated network environment as that depicted in Fig. 3.3 is considered.

First, the data file size is varied from 1 Gbits up to 10 Gbits, and the impact of this variation on the transfer delay, i.e. the time (in seconds) required to transfer the corresponding file chunks to the cloud storage while maintaining the same number of UEs and BSs, is investigated. The results are captured in Fig. 3.8.

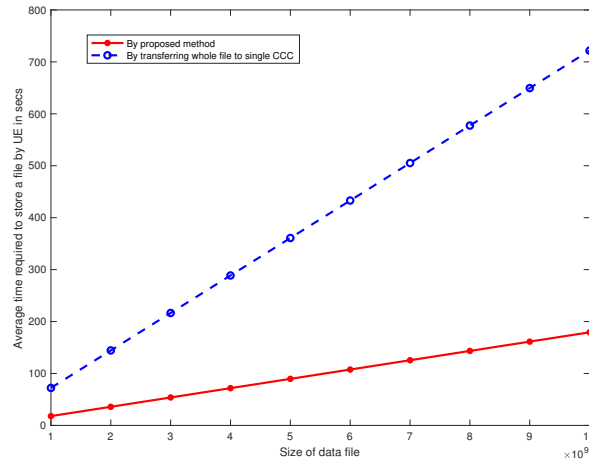


Figure 3.8: Data transfer delay vs. data file size

In Fig. 3.8, as expected, it is observed that when the size of the data file increases, the time required to transfer the file also increases; and this increase is linear in the case of the conventional method. In contrast, for the transfer of the same file, our proposed algorithm yields a significant reduced transfer time, i.e. almost 1/4 times of the time taken by the conventional method for a file of size 10 Gbits.

Second, the number of UEs in the network is varied from 11 to 20 and the impact of this variation on the transfer delay of a 1 Gbits data file size to the cloud storage is investigated. The results are captured in Fig. 3.9.

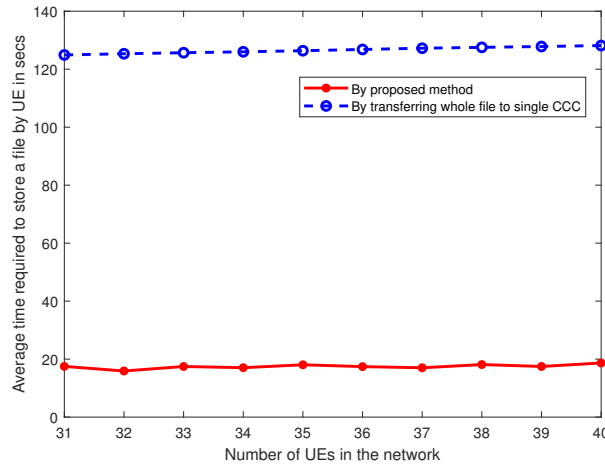


Figure 3.9: Data transfer delay vs number of UEs

In Fig. 3.9, it is observed that under the conventional method, the time required to transfer the file increases linearly with the number of UEs. However, the performance of our proposed algorithm is insensitive to the number of UEs, but slightly varies with the locations of UEs.

Third, the number of femto BSs in the network is varied from 3 to 10 and the impact of this variation on the transfer delay of a 1 Gbits data file size to the cloud storage is studied. The results are captured in Fig. 3.10.

In Fig. 3.10., it can be inferred that the performance of our proposed algorithm significantly improves that of the conventional method when the number of femto BS in the network is increased. When the number of femto BSs is increased from 3 to 5, the time required by our proposed algorithm to transfer the 1 Gbits data file is reduced by about 5 seconds compared to that taken by the conventional method. This reduction is even more pronounced when the number of femto BSs continues to increase.

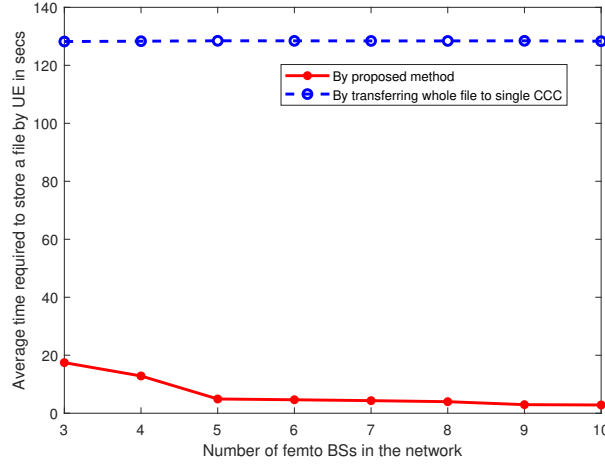


Figure 3.10: Data transfer delay vs number of femto BSs

3.5 Application of the Proposed RAID Based Algorithm for Big Video Data Files Transfer Over HetNet in 5G Systems

In this section, the above mentioned RAID based algorithm (Case of Data Chunk Size Proportional to Uplink Capacity) is applied for the transfer of big video data files. The same aforementioned network model (Fig. 3.1) is considered, where N_i^{paths} , the number of paths that the i^{th} UE can use to reach the cloud data center is also twice the number N_i^{UEs} of the BSs associated with that particular UE.

3.5.1 Big Video data File Preprocessing

A big video data file to be transferred to the cloud storage is compressed using the well-known High Efficiency Video Coding (HEVC) (or H.265) H.265 video coding algorithm [14], which compresses it to a smaller sized file. The compressed video file is split into N_i^{chunks} , where the size d_i^j of the i^{th} chunk to be transmitted on

the data link passing through the j^{th} BS is calculated [51] as:

$$N_i^{paths} = 2 \cdot N_i^{UEs} \quad (3.21)$$

$$N_i^{chunks} = N_i^{paths} - 2 \quad (3.22)$$

$$d_i^j = \frac{D \cdot (\beta_i^j - \beta_i^{min})}{\beta_i^{sum} \cdot (\beta_i^{max} - \beta_i^{min})}. \quad (3.23)$$

Here, the right hand size of Equation (3.2) is justified by the fact that in provision to the design of a data recovery technique that can protect against a failure to retrieve the data chunks in the cloud, two of the total available paths are reserved for the transmission of the associated syndromes.

3.5.2 Proposed Big Video Data File Transmission Algorithm

Assume that the pre-processed compressed video file is ready for transmission to the cloud data center through the available multiple data paths. Each of these paths has a varied maximum data rate calculated using Equation (3.9) following the Shannon-Hartley Theorem [35]. The compressed file is then spilt into $N_{chunks} = N_{data-paths} - 2$ data chunks. Then, these data chunks along with the data recovery syndromes ϕ and φ are transmitted to the cloud data center through the available data paths. There exists N_{UE} active UEs in the network and each i^{th} UE has J_i set of data paths available for data transmission. The goal of our proposed algorithm is to determine the size of the data chunks to be transmitted on each data path $D = \{d_1, \dots, d_{N_{chunks}}\}$ and the number of PRBs allocated to each data path $Z = \{\zeta_1, \dots, \zeta_{N_{chunks}}\}$. Our algorithm is meant to choose the data chunk sizes and PRB allocation such that the average of the delays experienced by all active UEs in the network is minimized. The delay experienced by a UE is the maximum of the delays experienced by all data paths. This problem can be

formalized as:

$$\{D, Z\} = \arg \min_{\tilde{D}, \tilde{Z}} \left\{ \frac{1}{N_{UE}} \sum_{i=1}^{N_{UE}} \max_{j \in J_i} DT_{i,j}^{total} \right\} \quad (3.24)$$

The major tasks of the proposed algorithm are as follows: (1) Decision on splitting of data file into multiple data chunks; and (2) Allocation of the uplink PRBs.

All available uplink PRBs are allocated among all the active UEs in the network with a reuse factor of 1. The number of PRBs allocated to a UE are equally allocated to all the associated BSs. Besides, a data file is spilt into a number of data chunks equal to the number of data paths available to the UE minus two. The size of each data chunk is directly proportional to the maximum data rate possible on the data path on which the corresponding data chunk is supposed to be transmitted. The flowchart of the proposed algorithm is given in Fig.3.11.

In the proposed algorithm, the size of the data chunk is proportional to the up-link capacity of the data link on which it is to be transmitted. For the calculation of the data recovery syndromes ϕ and φ , it is considered that the size of ϕ (resp. φ) is that of the largest data chunk (resp. the smaller data chunks), padded with zeros using Equation (3.17). As earlier, these syndromes are obtained using the aforementioned Equations (3.18-3.19), then they are saved to the cloud storage, along with the data chunks of the file. Two of the available data paths are reserved for the transmission of these syndromes, and the calculation of the lost data chunks can be found in [50].

3.5.3 Simulation Results: RAID Based Algorithm for Big Video Data Files Transfer

In this section, the simulation results showing the performance of the proposed RAID based algorithm in the case of big video data files is presented. The same simulated network environment as that depicted in Fig. 3.3 is considered. A num-

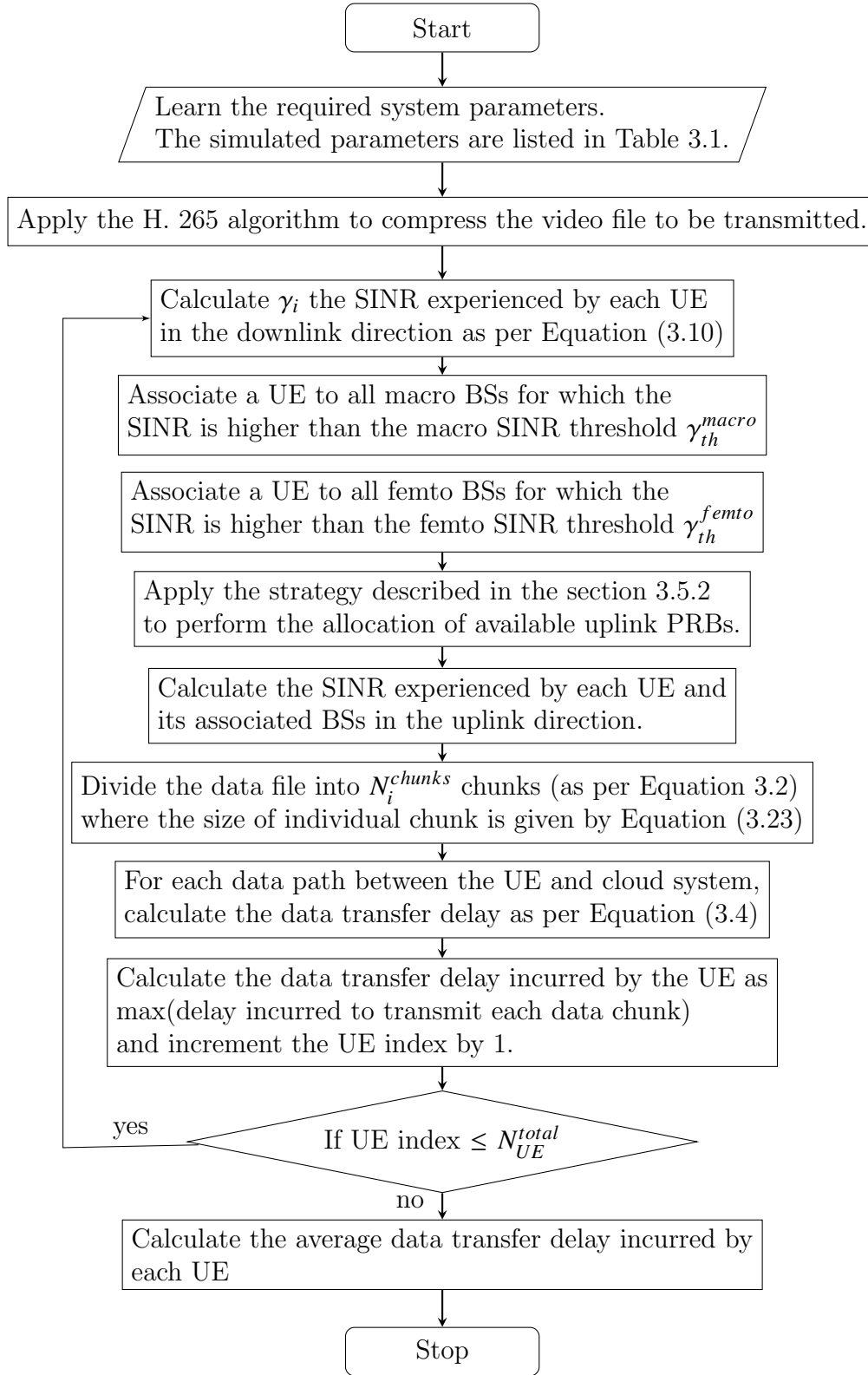


Figure 3.11: RAID-based algorithm for Big video data files transfer

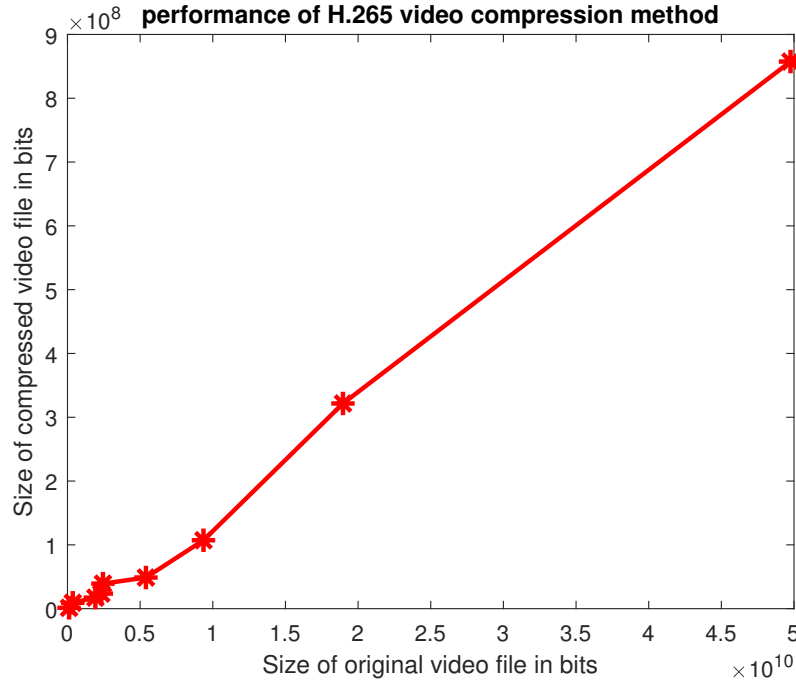


Figure 3.12: Performance of video compression algorithm.

ber of multiple video files are coded using the HEVC algorithm and the compressed video files are transmitted on the simulated network.

First, the performance of the HEVC algorithm considering the size of the input and output files is depicted in Fig. 3.12. In Fig. 3.12, it can be observed that with the application of the HEVC algorithm, the file size reduces significantly with a varying factor ranging from 40 to 50 depending on the content of the video file. It can be noted that three different input files with size around 20 Gbits have generated output files with varying sizes. The reason is that video files with highly correlated frames will be compressed efficiently compared to the ones with low correlation among the frames.

Second, the data file size is increased and the impact of this variation on the transfer delay, i.e. the time (in seconds) required to transfer the corresponding data file size while maintaining the same number of UEs and BSs is investigated. The results are captured in Fig. 3.13.

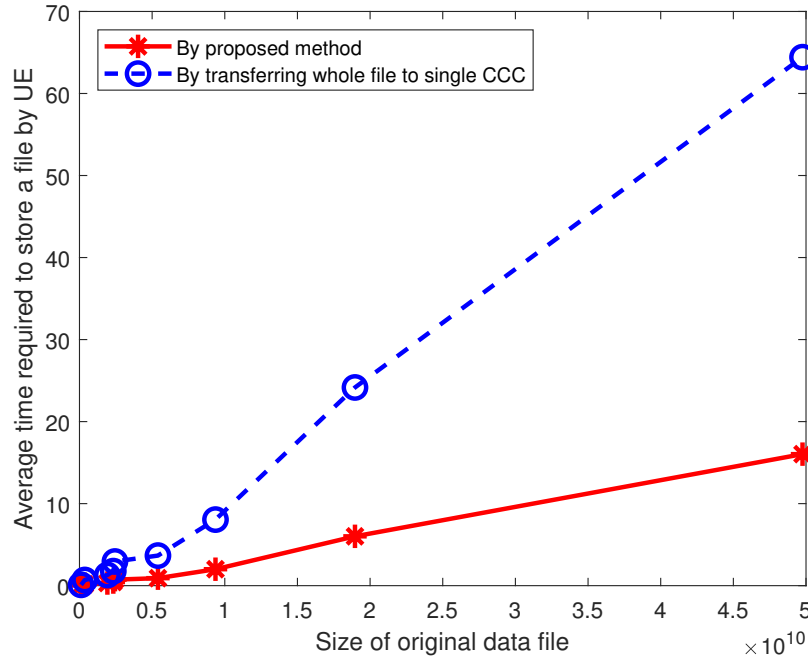


Figure 3.13: Impact of the variation in data file size on the data transmission delay.

In Fig. 3.13, as expected, when the size of the video file increases, the time required to transfer the file also increases linearly for both the proposed and conventional methods. The rate of increase in propagation delay with increase in the size of data file is lower with proposed method than that of conventional method, proving that the proposed algorithm outperforms the conventional one.

Third, the number of UEs in the network is varied from 11 to 20, and the impact of this variation on the transmission delay during the transmission of 1.92 Gbits data file size to the cloud storage while maintaining the same number and locations of macro/femto BSs in the network is investigated. The results are captured in Fig. 3.14.

In Fig. 3.14, it can be observed that under the conventional method, the time required to transfer the file increases linearly with the number of UEs. However, the transmission delay under our proposed algorithm is insensitive to the number

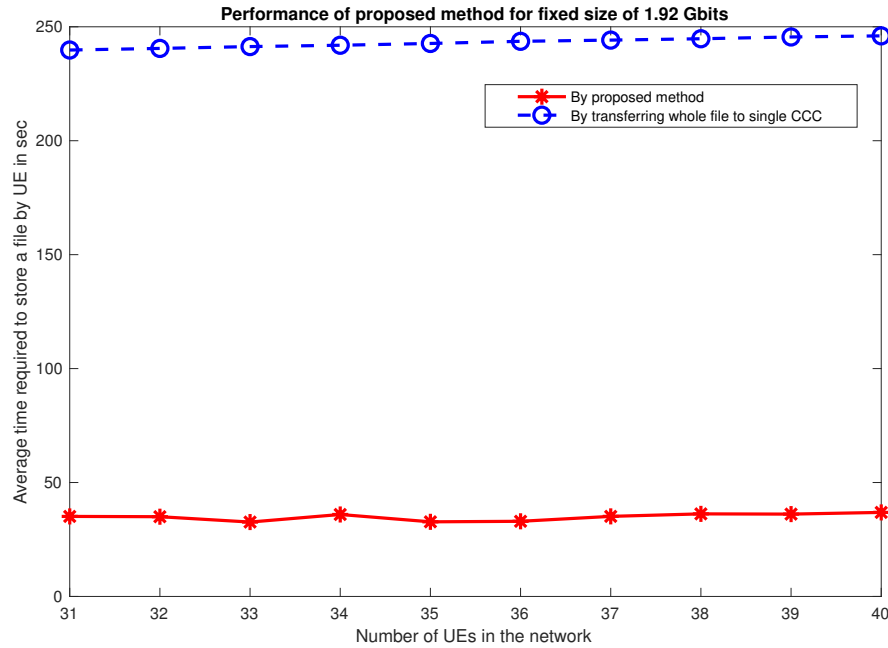


Figure 3.14: Impact of the variation in the number of active UEs in the network on the data transmission delay.

of UEs active in the network, but slightly varies with the locations of the UEs.

Fourth, the number of femto BSs in the network is varied from 3 to 10 and the impact of this variation on the transfer delay of a 1.92 Gbits data file size to the cloud storage while maintaining the same aforementioned HetNet setting is investigated. The results are captured in Fig. 3.15.

in Fig. 3.15, it can be observed that our proposed algorithm significantly improves when the number of femto BS in the network is increased. Also, the transmission delay performance of the conventional method does not yield any improvement. When the number of femto BSs is increased from 3 to 5, the time required by our proposed algorithm to transfer the data file is reduced by about 30 seconds, which is lesser than the one taken by the conventional method. This proves that the proposed algorithm efficiently utilizes the enhanced infrastructure provided by the HetNet.

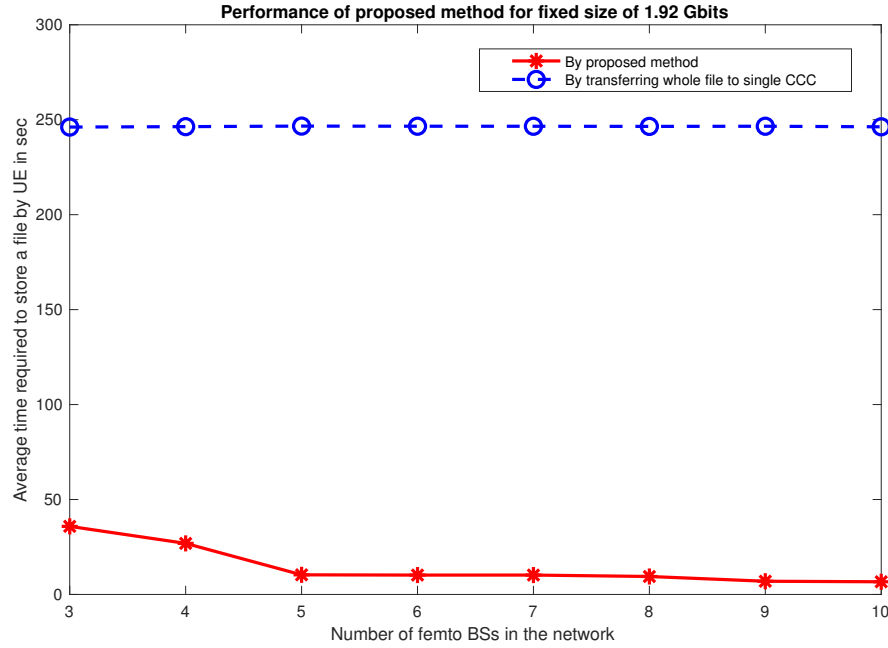


Figure 3.15: Impact of variation in number of small BSs on the data transmission delay.

3.5.4 Potential Drawbacks of Proposed Approach

The proposed solution approach is applicable only to a dense HetNet, where a UE has minimum four data-paths available to reach cloud storage system. Since, two data-paths are needed to transmit lost data recovery syndromes, minimum two additional data-paths are required to split a data file into two chunks and transmit them simultaneously to implement the solution. Additionally, the implementation of solution is characterized by additional complexity and overhead due to data splitting and calculation of lost data recovery syndromes. Moreover, bandwidth and storage overhead is associated with transmission and storage of lost data recovery syndromes.

Chapter 4

Mobile Cloud Storage over 5G: A Mechanism Design Approach

4.1 Introduction

In Chapter 3, we have addressed the problem of big data transmission over 5G HetNet to mobile cloud storage, where the goal is to significantly reduce the data transmission delay. The adopted solution has consisted in dividing the big data file into multiple data chunks, then concurrently transmitting them over the available data paths, along with two data correction syndromes.

This Chapter is an extension of the work carried in Chapter 3, where the multiple users scenario is considered, along with a mechanism design driven resource management approach. In this multiple users's scenario setting, each user has multiple transmissions opportunities provided by the overlapping coverage of the dense 5G HetNet to simultaneous transfer its big data through multiple wireless links. In order to prevent an abusive user from selfishly consume the wireless and infrastructure resources, a mechanism design framework is proposed to control the allocation of these resources among the multiple users. It is worth mentioning that mechanism design is a branch of game theory that merges computer science and economy and leads to a solution that exhibits several desirable

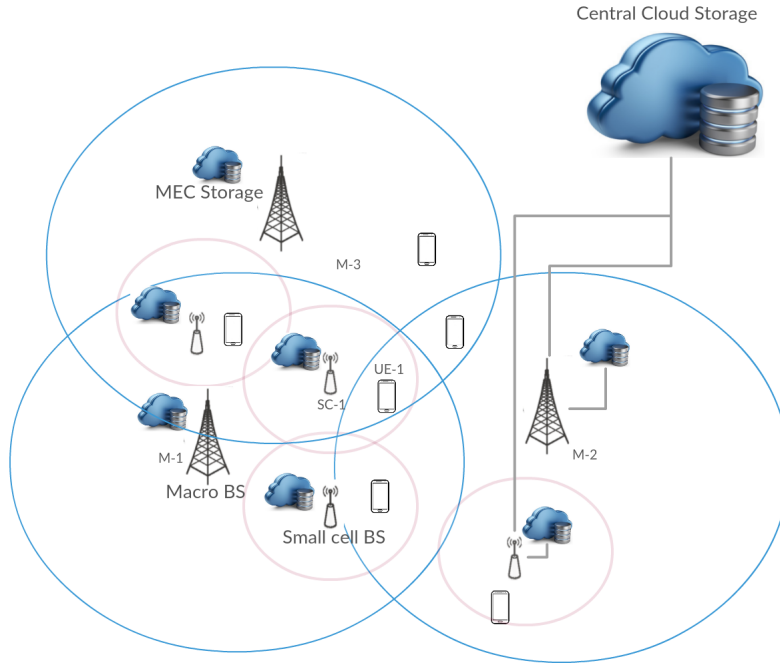


Figure 4.1: System Model.

features such as incentive compatibility, efficiency, and truthfulness.

In this Chapter, the system model depicted in Fig. 4.1, which includes a dense 5G HetNet enhanced with MCC and MEC technologies, is considered. In this model, each BS is equipped with a MEC storage center and a back-haul connection, which enables each BS to access the public cloud system. It is assumed that there are N mobile devices (UEs), each of which has access to multiple BSs. The network is designed following the 5G standards [34], [52] and details on the system parameters and the considered cell range expansion (CRE) technique [18] utilized to increase the usability of small cell BSs, have been provided earlier in Subsection 2.1.4 of Chapter 2. Without loss of generality, it is also assumed that the network condition does not change during a single storage execution.

Based on the above system model, a novel mechanism design based algorithm is proposed [15], which efficiently transfers the big data files with minimum transmission delay. In doing so, the overlapping nature of the dense 5G HetNet

is exploited to simultaneously transmit the mobile big data from the devices UEs to the cloud data centers. More precisely, the radio resources (here PRBs) and infrastructure resource allocation problem is formulated as a mechanism design problem; and a mechanism design based heuristic is proposed to solve this problem, which consists of a PRB/bandwidth allocation scheme presented as a graph colouring problem (GCP); a pricing algorithm, and a RAID based data transmission management scheme involving the use of the lost data recovery technique (introduced in Chapter 3) implemented at the individual UE level.

It should be noted that for the PRB/bandwidth allocation scheme, the considered relevant network condition parameters, i.e. number of UEs and set of UE pairs with mutual interference lower than the interference threshold, are transformed into an undirected graph, which helps in making the decisions on reusing the radio resources, the goal being to maximize the network performance. On the other hand, the pricing algorithm calculates the price a UE has to pay for the radio resources that it receives from the radio resource management algorithm. It is designed to be socially fair and to assign the rewards in such a way that the coalition is successful and sustainable. Finally, using our developed network simulator (introduced in Chapter 2), we have developed a Greedy algorithm and a Data Split Multiple UE (DSMU) algorithm and used these as benchmarks to evaluate the effectiveness of the proposed heuristic in terms of data transmission delay, response of the algorithms to increased service load (i.e. increase in the number of UEs) and response of the algorithms to an enhancement in the network infrastructure (i.e. increase in the number of small BSs). In addition, a mathematical analysis of the proposed algorithm to demonstrate its desirable properties from a game theoretical perspective such as incentive compatibility, efficiency and individual rationality, is proposed. Finally, simulation results of the mechanism design based method are presented.

4.2 Preamble

4.2.1 Multiple Data Paths Available to a UE

In the proposed system model, the BS association decision with a UE depends on the SINR experienced by the user on the downlink transmission link, knowing that the UE can be associated to multiple BSs. Typically, a UE will be associated to only those BSs from which it has experienced a SINR higher than a prescribed SINR-threshold value set by the network designer. Moreover, as per the cell range expansion technique [18], the SINR threshold γ_{femto} for the association decision with a femto BS is obtained as $\gamma_{femto} = \gamma_{macro} - \alpha$, where α is the cell selection bias and γ_{macro} is the SINR threshold for the BS association decision with a macro BS. Hence, at any instant t , a UE can be associated with multiple BSs on the condition that each of these BSs provides a SINR higher than the respective $\gamma_{femto}/\gamma_{macro}$ value. Due to the fact that a UE can be associated with multiple BSs, multiple paths (also referred to as datalinks) between a UE and the cloud storage system may be available for data transfer purpose. For instance, UE-1 in Fig. 4.1 can follow eight different routes that are available to reach the cloud storage, which are:

- UE-1 \Rightarrow SC-1 \Rightarrow MEC storage
- UE-1 \Rightarrow SC-1 \Rightarrow central cloud storage
- UE-1 \Rightarrow M-2 \Rightarrow MEC storage
- UE-1 \Rightarrow M-2 \Rightarrow central cloud storage
- UE-1 \Rightarrow M-1 \Rightarrow MEC storage
- UE-1 \Rightarrow M-1 \Rightarrow central cloud storage
- UE-1 \Rightarrow M-3 \Rightarrow MEC storage
- UE-1 \Rightarrow M-3 \Rightarrow central cloud storage

4.2.2 Mechanism Design

Mechanism design is a sub-field of game theory and microeconomics which has been used to model, analyze and solve many design problems in distributed environment, where participants cannot assumed to follow an algorithm, but their own self-interest [53]. The participants who are referred to as agents are capable of manipulating the algorithm, therefore, the designer of the algorithm should ensure that the interests of the agents are best served by following the designer's expectations. Agents are rational or selfish in the game theoretic sense of making the decisions consistently in pursuit of their own individual goals. It is assumed that each agent is intelligent enough to know everything about the underlying game that a game theorist knows and each agent can make similar inferences about the game that a game theorist can make. Mechanism design can be viewed as reverse engineering of game theory where the rules of a game are designed by the game theorist to achieve a specific desired outcome.

A typical mechanism design optimization problem consists of the following elements [53]:

- A game is played among N selfish, intelligent and strategic agents, and each of these agents has some private information (denoted as type (\hat{t}_i)). The rest of the game information is a public knowledge.
- Each agent i declares its type t_i to the mechanism which may not be same as its true type \hat{t}_i .
- The *output specification* is an objective function $g(O, T)$ that maps the type vector $T = \{t_1, \dots, t_n\}$ to an output vector $O = \{o_1, \dots, o_n\}$. The goal of the mechanism is to find an output vector that minimizes the function $g(O, T)$.
- The *valuation* v_i is a quantification of the i^{th} agent's value from the output o_i .

- Along with the output, a mechanism hands in a payment value p_i to the agent, calculated by using a payment function $p(t_i, O)$. Consequently, the agent's utility is determined as $u_i = v_i(o, t_i) - p_i$. The mechanism designer's aim is to design an output vector and payment function which maximizes the utility of each agent.

4.3 Proposed RAID based Data Transmission Management Scheme

The major constraint on the successful functioning of the transmission technique proposed in [11] is the limited uplink radio resources. As per the LTE-A standards [34], the up link spectrum offers 120 PRBs which needs to be shared among all active UEs. In the above mentioned network scenario, due to the overlapping coverage of multiple BSs, the majority of the active UEs have multiple data-paths available to transmit the data chunks simultaneously. However, the number of PRBs available to these data links for the uplink data transmission are finite. Therefore, an intelligent allocation and reuse of the available PRBs to these data paths is necessary in order to achieve the best possible network performance.

On the other hand, the problem addressed in the proposed radio resource management (RRM) scheme is to allocate limited PRBs to all active UEs in the network, where each UE has a different size of data file to transfer. Moreover, each UE has a different number of data links available and may experienced a varied SINR.

In summary, the procedural details of the proposed solution comprising a data management and lost data recovery (DMLDR) scheme and radio resource management (RRM) scheme are given in Fig. 4.6.

In our proposed solution framework, multiple MEC servers are expected to share a distributed storage, therefore fault tolerance should be taken into account [54] since it is likely that any of these devices may fail. Moreover, the data

chunks are transmitted over erroneous wireless links, which may corrupt the data while transmission. With high probability of error in the data retrieval, the data recovery is critical. In this Chapter, the same Lost Data Recovery Technique introduced in the design of the RAID based algorithm for faster data transfer: Case of Data Chunk Size Proportional to Uplink Capacity (in Chapter 3) is adopted, where it is assumed that the size of the data chunk is proportional to the uplink capacity of the data link on which it is to be transmitted. It should be noted that using the available data links, these data chunks are transmitted to the cloud data storage along with the calculated data recovery syndromes.

4.4 Proposed Radio Resource Management Scheme

4.4.1 PRB Allocation Problem

In the considered scenario, the PRB allocation problem is formulated as a mechanism design problem as follows:

- A game is played among N selfish, intelligent and strategic agents who are active UEs in the network, and each active UE has a data file that it needs to store in the cloud system; hence it has a type $\hat{t}_i = D_i \geq 0$, where D_i is the size of the data file.
- Each UE declares a type t_i which might not be equal to its true type \hat{t}_i .
- The output function $g(O, T)$ is defined as

$$g(O, T) = \frac{1}{N} \cdot \sum_{i=1}^N \frac{1}{\lambda_i} \quad (4.1)$$

$$\lambda_i = \frac{t_i}{\beta_i} \quad (4.2)$$

where O is the PRB allocation vector assigned by the mechanism, λ_i is the transmission delay experienced by the i^{th} UE while transmitting a data file of

size t_i , and β_i is the maximum data rate assigned to the i^{th} UE transmission.

- The valuation $v_i = \frac{1}{\lambda_i} = \frac{t_i}{\beta_i}$ is the reciprocal of the delay experienced by the i^{th} UE. The aim of the mechanism is to discover the O (PRB allocation vector) that minimizes the output function $g(O, T)$.
- In addition to the PRB allocation, the mechanism assigns the payment value p_i to the UE, which represents the price that the i^{th} UE needs to pay to the CSP. This value is directly proportional to the amount of PRBs assigned to it and the total amount of interference that the UE causes to the rest of the UEs in network.
- A part of public knowledge is the interference graph representation of the current network condition (as described in Section 4.4.2.1).

4.4.2 Proposed Algorithmic Solution to the PRB Allocation Problem

4.4.2.1 PRB Allocation Problem as GCP:

Once the mechanism have acquire all the required knowledge regarding the network scenario, it transforms the bandwidth (PRB) allocation problem in the considered network into a Graph Colouring Problem (GCP), then it generates the interference that will be used by the proposed allocation algorithm 2. Following the technique proposed in [32], the PRB allocation problem can be represented as a k -graph coloring problem, where k is the number of available PRBs. In the graphs depicted in Fig. 4.3, Fig. 4.4 and Fig. 4.5, $G = (V, E)$, where V is the set of vertices and E is the set of edges. The k -colouring of G is a function

$$c: V \rightarrow \{1, 2, \dots, k\}$$

such that $c(i) \neq c(j)$, where E contains an edge (v_i, v_j) connecting the vertices $v(i)$ and $v(j)$. In order to formulate the PRB allocation problem as a graph colouring problem, the (UE-BS) are represented as vertices and a PRB is represented as a colour. From now on, the terms data-link and vertex will be used interchangeably as well the terms colour and PRB. In addition, the minimum SINR performance expected at the user terminals is set to γ_{thresh} . The corresponding interference threshold (I_{thresh}) can be calculated as $I_{thresh} = \min(\hat{p}_1, \hat{p}_2) - \gamma_{thresh}$ (in dB), where \hat{p}_1 and \hat{p}_2 are the powers received by data-link1 and data-link2 respectively. The presence of an edge is restricted to those users whose mutual interference is higher than I_{thresh} . The presence of an edge connecting two users signifies that the users, if allocated the same PRB, will cause a mutual interference that is higher than I_{thresh} . The network scenario depicted in Fig. 4.2 are represented by the graphs shown in Figs.4.3-4.5, where γ_{thresh} is set to the values 20dB, -10dB and -40dB respectively.

The PRB allocation problem is transformed into a graph colouring problem. For a particular instance of the network condition, the number and position of the vertices in the graph remain constant, but the number of edges varies with the value of the SINR threshold (γ_{thresh}). When γ_{thresh} increases, it is clear that I_{thresh} decreases since $I_{thresh} = \min(\hat{p}_1, \hat{p}_2) - \gamma_{thresh}$ [in dB], where \hat{p}_1 is the power received by UE-1 and \hat{p}_2 is the power received by UE-2. An edge is present between two vertices if the mutual interference is higher than I_{thresh} ; hence the number of edges increases with an increase in γ_{thresh} as inferred to by Figs.4.3 - 4.5. The higher the number of edges, the higher the number of constraints on the PRB allocation, hence the complexity of PRB allocation problem increases with an increase in the value of the SINR threshold.

4.4.2.2 Proposed PRB Allocation and Pricing Algorithm:

Mechanism design is a methodology of designing the rules of a game to achieve the desired goal of the game. As a matter of fact, the Vickrey-Clarke-Groves

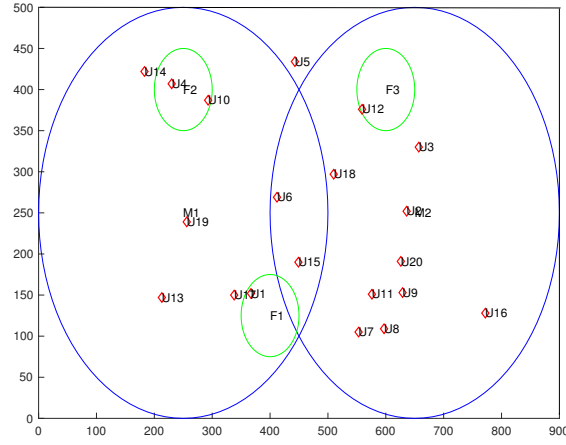


Figure 4.2: Simulated network scenario.

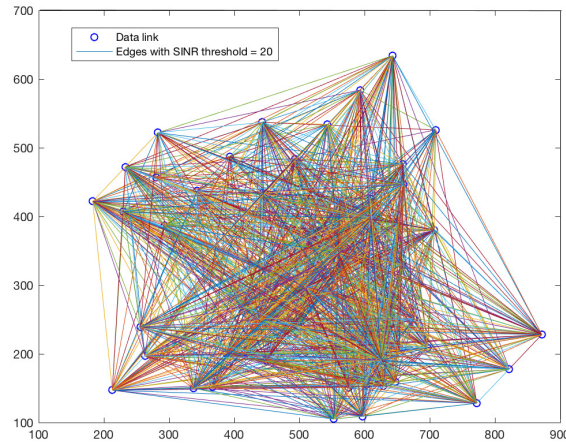


Figure 4.3: Interference graph with $\gamma_{thresh} = 20\text{dB}$.

(VCG) mechanism has been widely used in multi-agent systems to devise an efficient resource allocation when the agents have private independent valuations [16]. This is because it exhibits several desirable properties, including incentive and compatibility. It also ensures that the resulting resource allocation is efficient (i.e. the social welfare is maximized) and individually rational (i.e. it ensures that all the participating players derives a non-negative utility from the mechanism) [16]. For this reason, we have considered a PRB allocation based on the VCG

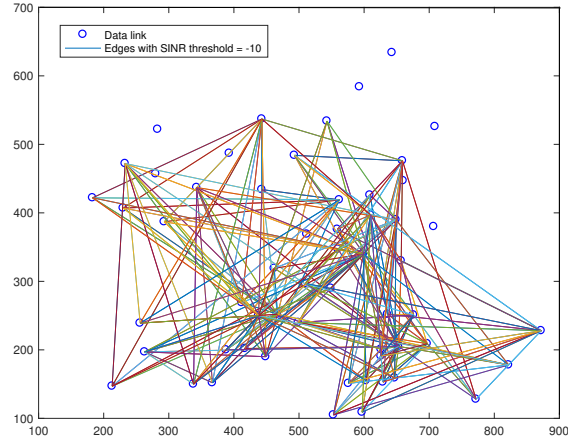


Figure 4.4: Interference graph with $\gamma_{thresh} = -10\text{dB}$.

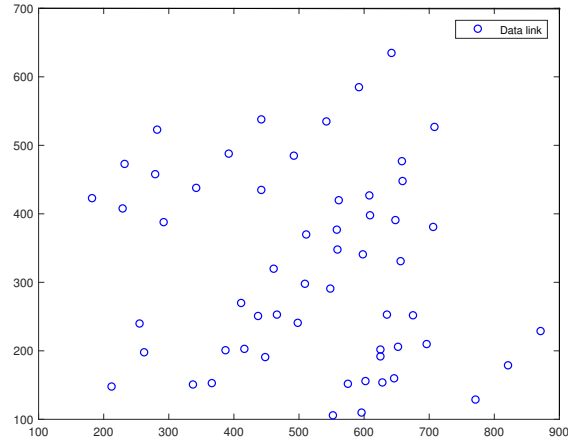


Figure 4.5: Interference graph with $\gamma_{thresh} = -40\text{dB}$.

mechanism. Here, the VCG mechanism is applied to the utilitarian mechanism design problems.

For any mechanism $m(O(t), h(O, t))$ to be classified as a VCG based family of mechanisms, it should exhibit the following properties [53]:

- The mechanism must be utilitarian.

$$O(t_i) \in \arg \max_{\check{O}} \left\{ \sum_{i=1}^n v_i(\cdot) \right\} \quad (4.3)$$

- It must define the payment functions for the agents as per the following equation:

$$p(t_i) = \sum_{j \neq i} v^j(t^j, O(t_i)) + h(t^{-1}), \quad (4.4)$$

where $h(t^{-1})$ is an arbitrary function of t^{-1} specific to the problem under consideration. The intuition behind the payment function of a VGC mechanism is that a player pays for its ‘social cost’ i.e. the damage it imposes on the society. The first term of Equation (4.4) corresponds to the optimal welfare for other players if the i^{th} player is not participating in the game, and the second term of Equation (4.4) corresponds to the welfare of other players with the current type T . Our proposed mechanism complies to these requirements.

Algorithm 2 PRB allocation algorithm

- 1: Determine the data-links and generate the interference graph following the details from the Section 4.4.2.1
 - 2: **procedure** PRB ALLOCATION($t_i, G = (V, E)$)
 - 3: Perform the initial assignment of PRBs to all UEs, where the number of PRBs assigned to a UE is proportional to the size of the data file that it needs to transmit i.e. $O^{initial} = \{o_1, \dots, o_n\}$ and $O_i^{initial} \propto t_i$.
 - 4: Split equally $O_i^{initial}$ among data-links available to the i^{th} UE.
 - 5: **for** all 2-tuple of data-links(k, j) in the network **do**
 - 6: **if** $E_{k,j} \neq 1$ (if edge is not present between the i^{th} UE and k^{th} UE) **then**
 - 7: $O_j = O_k^{initial} \cup O_j^{initial}$
 - 8: $O_j = O_k^{initial} \cup O_j^{initial}$ where $E_{k,j}$ is equal to 1 if an edge exists between the k^{th} UE and the j^{th} UE and 0 otherwise.
 - 9: **end if**
 - 10: **end for**
 - 11: **end procedure**
-

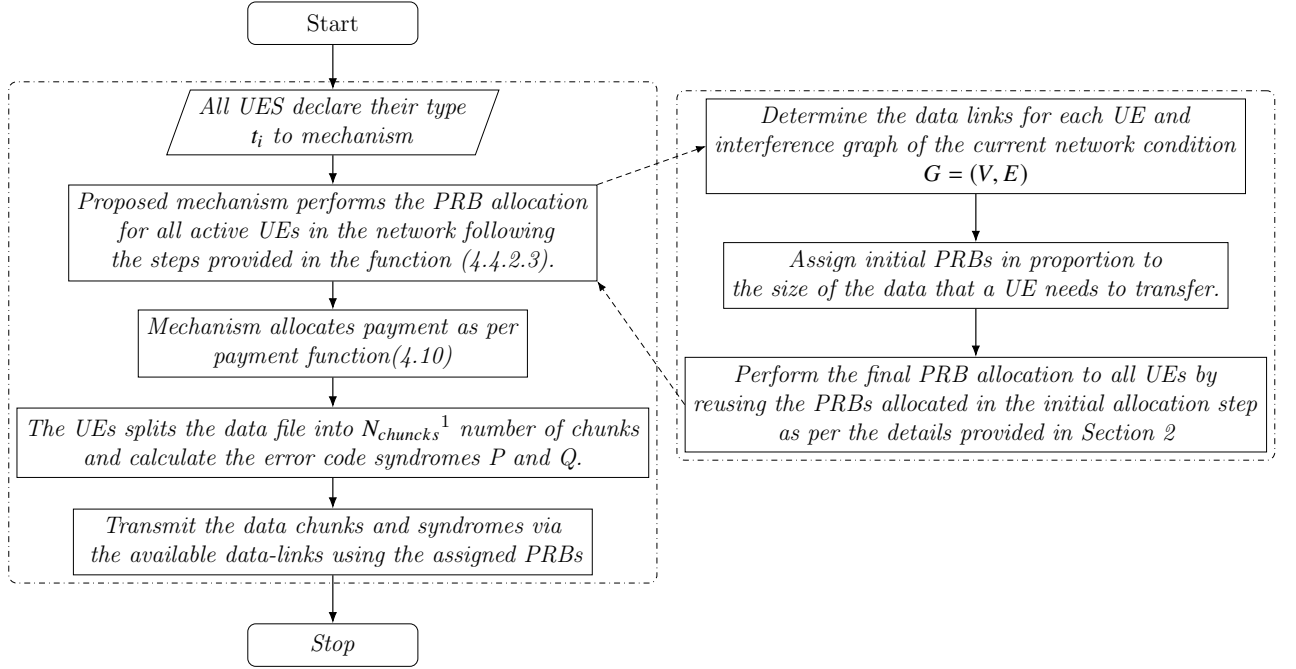


Figure 4.6: Procedural details of the proposed solution comprising of DMLDR and RRM schemes.

4.4.2.3 Output Function:

The goal of our proposed framework is to minimize the average delay experienced by all active UEs in the network. Once all UEs declare their types t_i (the size of file to be transmitted) to the mechanism, the optimization problem can be defined (as per (4.6)) and the output (i.e. the PRB allocation) is determined. The optimization problem can be described as follows:

$$O = \arg \max_{\check{O}} \{v_i(\check{O})\} \quad (4.5)$$

$$O = \arg \max_{\check{O}} \left\{ \frac{1}{N} \sum_{i=1}^N \frac{t_i}{\beta_i(\check{O})} \right\} \quad (4.6)$$

$$\beta_i = W_i \cdot \log_2 \left(\frac{\hat{P}_i^J}{\tau_i \cdot [1 + (\eta/100)]} \right) \quad (4.7)$$

$$\tau_i = \sum_{j=1, j \in o_i, j \neq i}^{N_{BS}^{total}} \hat{P}_i^j \quad (4.8)$$

where \check{O} is a set of solutions representing the PRB allocation to the requesting UEs. $O = \{o_1, \dots, o_N\}$, o_i is a set of PRBs assigned to the i^{th} UE, N_{BS}^{total} is the number of BSs in the network, W_i is the amount of bandwidth available to the i^{th} UE, and \hat{P}_i^j is the power received by the i^{th} UE from the j^{th} BS.

4.4.2.4 Payment Function:

Once the PRB allocation has been formulated, the payment i.e. the price a UE needs to pay for the PRBs it would be using for data transfer purpose is calculated by the payment function $p(t_i)$ (4.10). Any mechanism can be stated to be efficient if it has an efficient allocation rule/output function. The VCG mechanism is an efficient mechanism which uses the payment function $p : T \Rightarrow \mathbb{R}$. As stated earlier, the payment function p_i is the difference between the optimal welfare for other players if the i^{th} player is not participating and the welfare of other players with the current type T . An intuition behind this payment function is that a player should pay for the damage that it imposes on the society [55].

Following the principles of VCG mechanism in the considered network scenario, an efficient payment function would be

$$p(t_i) \propto \overline{\lambda_j^1} - \overline{\lambda_j}, \quad (4.9)$$

where $\overline{\lambda_j^1}$ is the average delay experienced by other UEs if the i^{th} UE disappears from the network; and $\overline{\lambda_j}$ is the average delay experienced by all UEs with all active UEs present with the same declared type T . The difference between these two terms is dependent upon two factors namely:

1. Additional remaining PRBs of the UEs would be received if the i^{th} UE disappears from the network.

2. There would be reduction in the interference received by other UEs if the UE disappears from the network.

Considering these factors, the payment function of our proposed method is obtained as:

$$p(t_i) = 2^{\hat{W}_i} + pw_1 \cdot \hat{\psi}_i + pw_2 - 1 \quad (4.10)$$

where the effect of the first factor is considered in the first term of Equation (4.10), where the payment value is directly proportional to the normalized amount of PRBs assigned to the UE (\hat{W}_i). The effect of the second factor is considered in the second term of Equation (4.10) where $\hat{\psi}_i$ is the normalized amount of total interference caused by the i^{th} UE to the rest of the active UEs in the network as a result of the same output function. The constants pw_1 and pw_2 are the weights of the payment function, which are decided by the MNOs as per their billing requirements.

4.5 Analysis of the Proposed mechanism design based Method

Our proposed method inherits the desirable economic properties of the VGC mechanism as follows.

4.5.1 Incentive Compatible in Ex-post Nash Equilibrium

Definition 1. A mechanism said to be incentive compatible in ex-post Nash equilibrium if the best response strategy for all players is to reveal its true type even if it has the complete information about other agent's types [55] and [16]. ■

Definition 2. According to [16], the utility of the i^{th} UE is defined as:

$$u_i = v_i - p(t_i) \quad (4.11)$$

$$v_i = \frac{W_i}{t_i} \cdot \log_2 \left(\frac{\hat{P}}{\tau_i (1 + 0.01 \cdot \eta)} \right) \quad (4.12)$$

The valuation a player/UE gets from the game is proportional to the delay experienced by a UE when transferring its big data file. In Equation (4.11), if we ignore the terms which are independent of the declared type (t_i), the utility of a UE is obtained as:

$$u_i = \frac{W_i}{t_i} \cdot \log_2 \left(\frac{\hat{P}}{\tau_i (1 + 0.01 \cdot \eta)} \right) - 2^{\hat{W}_i} - \hat{\psi}_i \quad (4.13)$$

■

Lemma 1. *The proposed mechanism composed of an output and a payment function is incentive compatible in ex-post Nash equilibrium.*

Proof. Every VCG mechanism is incentive compatible [53]. With the close examination of (4.13), the interference caused to another active UEs (ψ_i) is proportional to the number of interfering UEs which are using at least one PRB similar to the PRBs assigned to the i^{th} UE. The number of UEs reusing the PRBs is proportional to the number of edges connected to a vertex in the network graph representing the UE. The interference graph is independent of the type (t_i) declared by the UEs. Similarly τ_i is independent of t_i , hence the utility of the UE reduces to the term, which depends on the UE's declared type.

$$u_i = \frac{W_i}{t_i} - 2^{\hat{W}_i} \quad (4.14)$$

Let \hat{t}_i be the true type i.e. the true data file size the i^{th} UE needs to be transfer. if $\hat{t}_i > t_i$ i.e. if a UE has declared a data file size larger than the real data

file size, more PRBs would be assigned to the UE. This would linearly increase the first term of Equation (4.14) while exponentially increasing the payment (second term of Equation (4.14)), hence the net utility will be reduced. On the other hand, if $\hat{t}_i < t_i$, i.e. a UE has declared a data file size lower than the real data file size, then less number of PRBs would be assigned to that UE, which is undesirable by that UE. \square

4.5.2 Efficiency

Definition 3. An allocation rule (or output function) of a mechanism is said to be efficient if it maximizes the social welfare [16].

$$O \in \arg \min_{\vec{O}} \forall_i \lambda_i \quad (4.15)$$

■

Lemma 2. *The proposed mechanism is efficient since it allocates the PRBs with the aim to maximize the average transmission delay experienced by each active UE in the network as per Equation (4.6).*

Proof. In the considered problem, the social welfare is equivalent to the inverse of the delay experienced. As the proposed mechanism is incentive compatible ex-post Nash equilibrium, it encourages the players (UEs) to declare their true types. Once each UE in the game has declared the true size of the data file it needs to transfer, the output function decides on the PRB allocation as well as the payment values for the UEs, which in turn maximizes the average value of the data rates experienced by all of active UEs in the network. \square

4.5.3 Individually Rational

Definition 4. Each player in the game is assumed to be 'intelligent'. The mechanism should encourage a selfish agent to join it rather than opt out of it by

ensuring that the utility derived by an agent within the mechanism is never less than that of the agents that are exterior to it. It is assumed that the utility that an agent gains from not joining the mechanism is 0. One needs to prove that the utility an UE gains in the mechanism is always ≥ 0 [16]. ■

Lemma 3. *The proposed mechanism is individually rational. The output/PRB allocation and payment functions have been designed in such a way that the mechanism encourages a selfish UE and participating MNOs to participate in the coalition.*

Proof. The proposed mechanism ensures that even with the worst network conditions, each participating UE receives at least 1 PRB which is equivalent to $W_i = 180\text{MHz}$. With this minimum possible amount of bandwidth, the proposed mechanism ensures that a UE can achieve a minimum possible utility (as per Equation (4.13)), which is always higher than the utility that a UE would achieve by not participating in the game. □

4.6 Simulation Results

The proposed algorithmic framework is executed on the simulated network model provided by MATLAB (see Section 2.1.4). A typical simulated network scenario is shown in Fig. 4.2. The goal of our framework is to efficiently transfer the big data files from mobile UEs to the cloud system with minimum data transmission delay. For comparison purpose, a Greedy algorithm and along with a Data Split Multiple UE (DSMU) algorithm are implemented and executed on the same network model. The performance of our proposed algorithm is compared against that of those algorithms.

4.6.1 Greedy Algorithm

In the implementation of the Greedy algorithm (GA), each UE associates itself to the BS with the maximum downlink SINR and the data file is transmitted to

the public cloud via this BS. The uplink PRB allocation in the GA algorithm is performed as follows: There are two macro BSs present in the simulated network, namely M1 and M2. The total available PRBs are equally assigned to all the UEs associated with either macro BS without any reuse. The available PRBs are again reused for assignment to the UEs associated to the femto BSs. On the other hand, the total PRBs are equally split among the UEs associated with any femto BS.

4.6.2 Data Split Multiple UE Algorithm

The DSMU algorithm is an enhanced version of a recently proposed algorithm [11], in which multiple active concurrent UEs have been considered. Typically, in this enhanced version, each UE has multiple data-links (i.e. wireless links between the UE and its associated BS) associated with it. All data-links from multiple UEs which are associated with macro BS M1 would be assigned the first half of the available PRBs and the rest of the PRBs would be assigned to the data-links associated with the macro BS M2. The whole set of PRBs is reused for assignment to the data-links which are associated with the femto BSs. The available PRBs are equally assigned to all data-links which are associated with any femto BS.

First, it is assumed that all active UEs in the network have a data file of equal size to transmit to the cloud. The data file size is varied from 1 Gbits to 10 Gbits and the impact of this variation on the average delay experienced by all UEs is investigated. The results are captured in Fig. 4.7. In Fig. 4.7, it is observed that the time required by our algorithm for complete file transfer is significantly lower compared to that generated by the greedy and DSMU algorithms.

Indeed, to transfer a data file of size 9 Gbits, the greedy and DSMU algorithms require approximately 10^8 secs while the proposed algorithm transfers the file in 10^4 secs. It can be observed that as the size of the data file increases, the time required to transfer it increases linearly in all cases.

Second, the number of active UEs in the network is varied and the impact

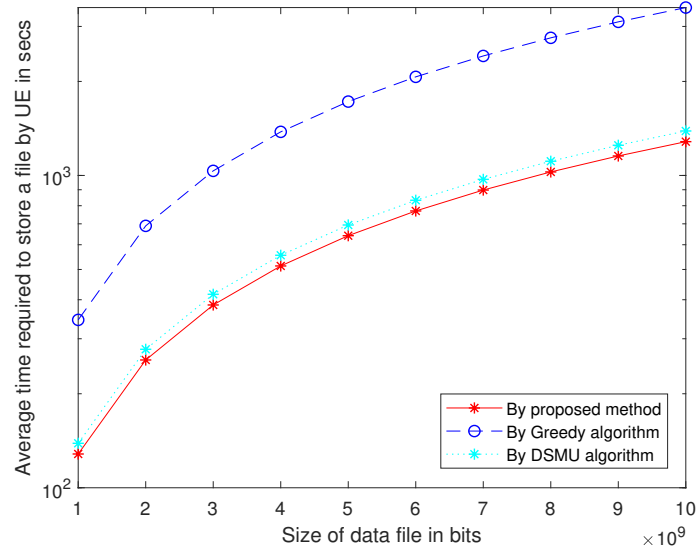


Figure 4.7: Time required by the UE to complete the transfer of the file vs. size of data file (on a logarithmic scale).

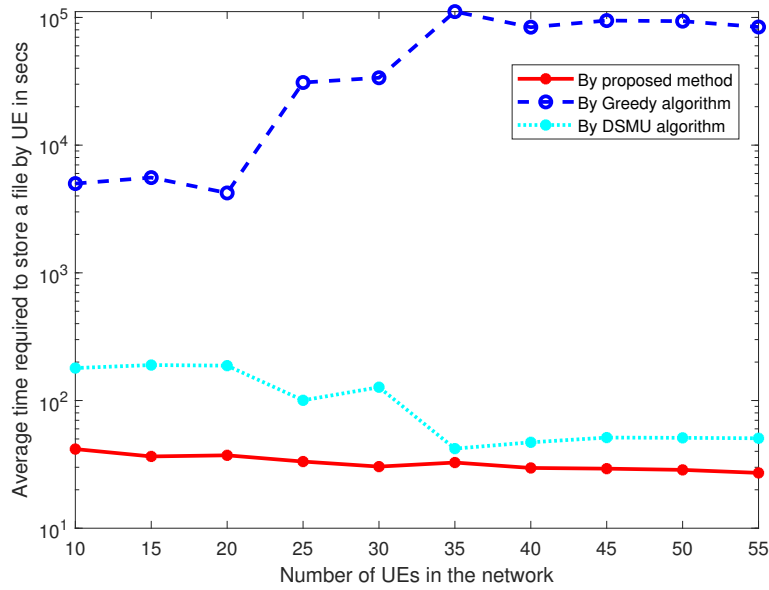


Figure 4.8: Time required by the UE to complete the transfer of the file vs. number of active UEs in the network (on a logarithmic scale).

of this variation on the average delay experienced by all UEs is investigated. The results are captured in Fig. 4.8. It can be observed that for the proposed scheme and DSMU algorithm, the average delay experienced by a UE is approximately 2.3×10^3 times less than that experienced by the greedy algorithm. It can be further observed that the proposed and DSMU algorithms are less sensitive to changes in the number of active UEs while the performance of the greedy algorithm fluctuates with the number and locations of the active UEs in the network. The rationale behind this stable performance of the proposed algorithm with the variations in the number and locations of the UEs is that the proposed algorithm utilizes multiple BSs covered by a UE and the PRB allocation is performed considering this data transmission strategy. Moreover, the PRB allocation algorithm also considers the network conditions, including the locations and number of active UEs in the network.

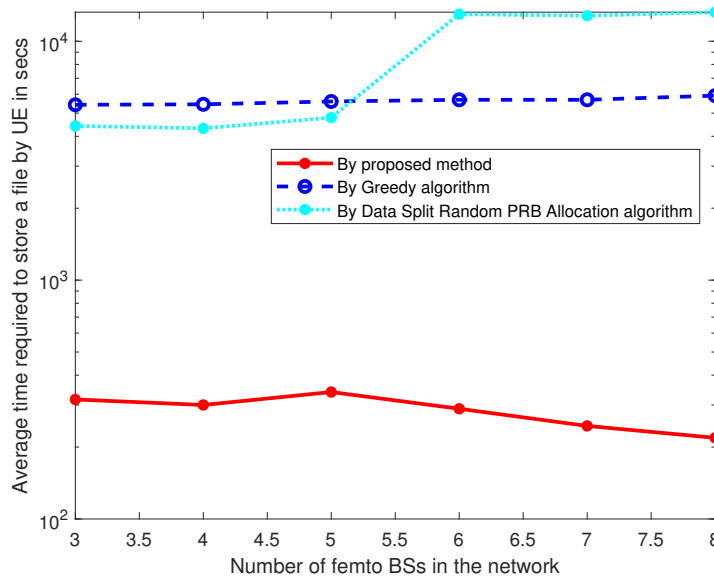


Figure 4.9: Time required by the UE to complete the transfer of the file vs. number of small cell BSs in the network.

Third, the number of small cell BSs in the network is increased and the impact of this increase on the average delay experienced by all UEs is investigated.

The results are captured in Fig. 4.9. It can be observed that the data transmission delay generated using our proposed algorithm is about 17 times lower than that generated by the greedy and DSMU algorithms. It can be further observed that as the HetNet becomes more denser, the delay experienced by the greedy algorithm remains constant, meaning that its performance deteriorates. On the other hand, with our proposed algorithm, the average delay experienced by a UE decreases. This is due to the fact that our proposed algorithmic framework exploits the intrinsic HetNet infrastructure and continues to improve the network performance when the number of small cell BSs is increased in the network.

Fourth, the heatmap from Fig.4.10 depicts the values of the proposed payment function under varying normalized number of PRBs allocated to a UE among all active UEs in the network and varying values of the normalized interference that the same UE causes to the rest of the UEs in the network. It can be observed that the payment value is directly proportional to the amount of PRBs allocated to a UE and the amount of interference a UE causes to the rest of the active UEs in the network.

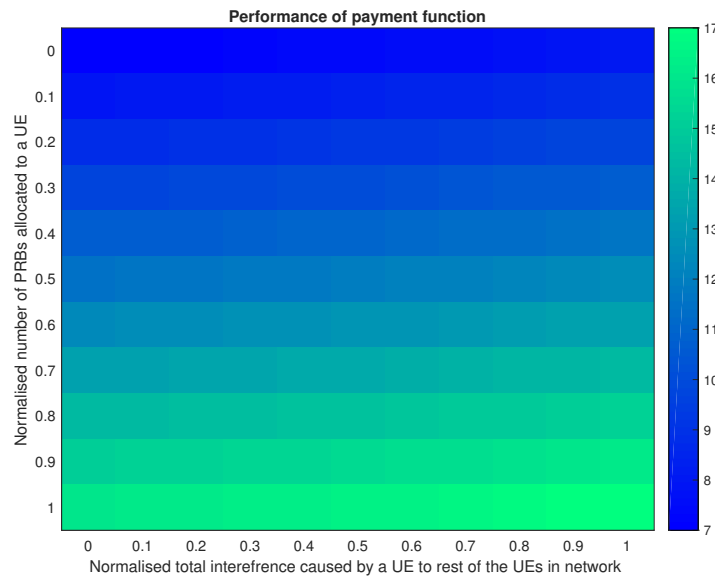


Figure 4.10: Payment function distribution.

4.6.3 Incentive Compatible ex-post Nash Equilibrium

To validate that our proposed algorithmic framework is incentive compatible ex-post Nash equilibrium, we have performed some simulations as follows. In the proposed framework, the type t_i is the size of data file that a UE needs to transfer to the cloud system. It is assumed that all UEs have the same data files of size 6 Gbits, and one of the active UEs has declared a data file size/type varying from 1Gbits to 10Gbits, but different from 6 Gbits (i.e. it is not the true type value). All other simulation parameters are inherited from Table II.

First, the declared type values is varied and the impact of this variation on the initial number of PRBs assigned to the UE under consideration without the application of frequency reuse is investigated. The results are captured in Fig. 4.11. It is observed that as the declared type by the UE increases, the number of PRBs assigned to the UE initially by the algorithm increases linearly.

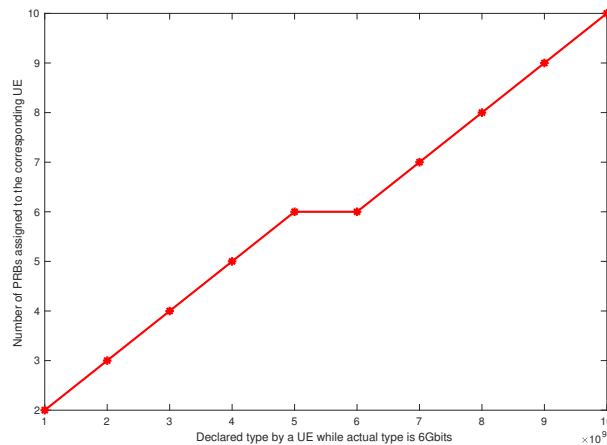


Figure 4.11: Initial number of PRBs assigned to a UE by the proposed algorithm for various declared type (t_i).

Second, the final number of PRBs assigned to a UE by the proposed algorithm when applying the frequency reuse technique is shown in Fig. 4.12. It can be observed that the initially assigned PRBs are reassigned to the UEs who do not share an edge with the current UE from the corresponding GCP. When the

declared type value is increased, the number of PRBs assigned to a UE slightly decreases. This is due to the fact that the frequency reuse factor is dependent on the interference graph, thereby on the network conditioning parameters such as locations of the UEs and base stations; and it is independent of the declared type values. The PRBs assigned to a UE by frequency reuse constitutes the majority of PRBs assigned to the UE. This is the rationale behind utilizing the amount of initial PRBs without frequency reuse (instead of the number of final PRBs) in the payment function.

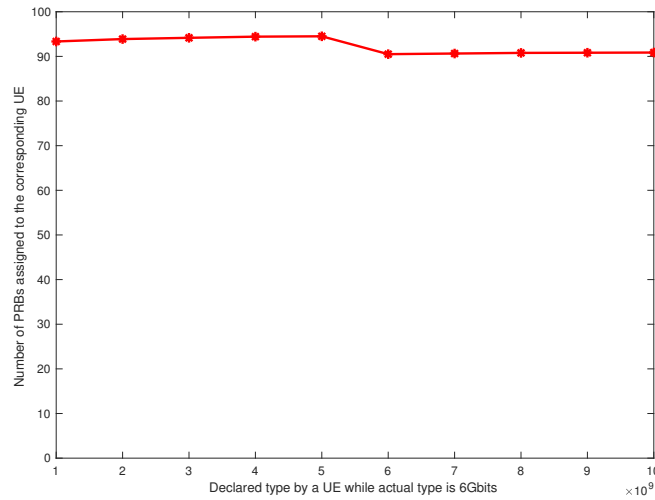


Figure 4.12: Final number of PRBs assigned to a UE by the proposed algorithm for various declared type (t_i) using frequency reuse.

Third, the number of PRBs assigned to a UE is directly proportional to the throughput that a UE can achieve. Indeed, the delay experienced by the UE under consideration when transferring its data file and the proportional payment value that it needs to pay for various declared types values, are depicted in Fig. 4.13. It can be observed that when the declared type value is increased, the delay experienced by the UE slightly decreases until it reaches its true value, then it suddenly increases, and then remains constant when the declared type values further increases. On the other hand, the payment value increases when

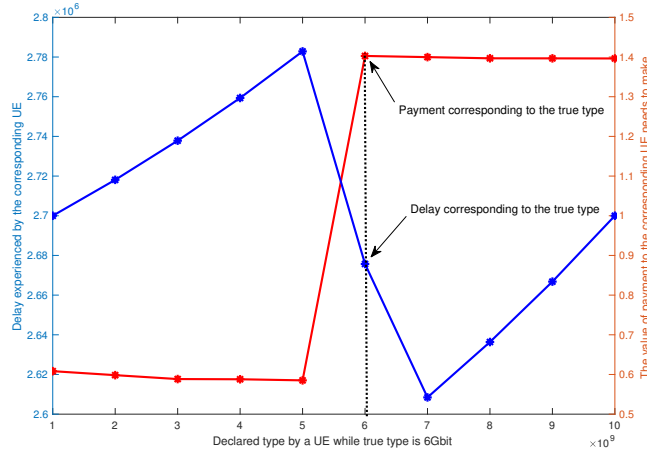


Figure 4.13: Delay experienced by the UE and the payment value a UE needs to pay for various values declared type (t_i) declared.

the declared type value increases and suddenly dips when the declared type is closer to its true value.

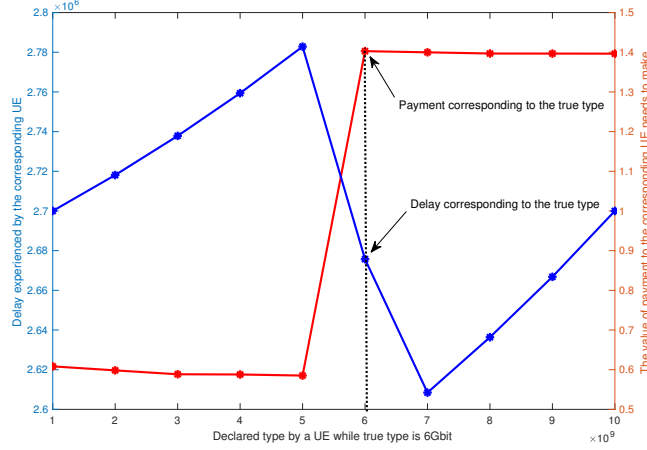


Figure 4.14: Net utility of a UE for various values of type (t_i) declared.

Fourth, the Net utility of the UE into consideration under varying declared type (t_i) values is investigated, where the Net utility of the UE is a function of the delay experienced by the UE in transferring the data file and the amount interference caused by the UE to the rest of the UEs in network. The results are

captured in Fig. 4.14. It can be observed that the utility of a UE is maximum (with experimental errors) when the declared data file size/type is closer to the true type size (6 Gbits). Therefore, the proposed mechanism is successful in encouraging a UE to declare its true type, i.e. it is incentive compatible ex-post Nash equilibrium.

Chapter 5

Joint Optimization of Radio and Infrastructure Resources for Energy Efficient Massive Data Storage in Mobile Cloud over 5G HetNet

5.1 Introduction

In Chapter 4, we have addressed the problem of radio resource and data transmission strategy in a dense HetNet equipped with MCC and MEC technologies; and proposed a mechanism design driven resource management scheme as solution approach. In this chapter, the same problem is addressed, and a different solution approach is proposed.

More precisely, the problem is formulated as a combinatorial optimization problem, and solved using an ACO based energy aware algorithm consisting of a data transmission management scheme, a PRBs/bandwidth allocation scheme.

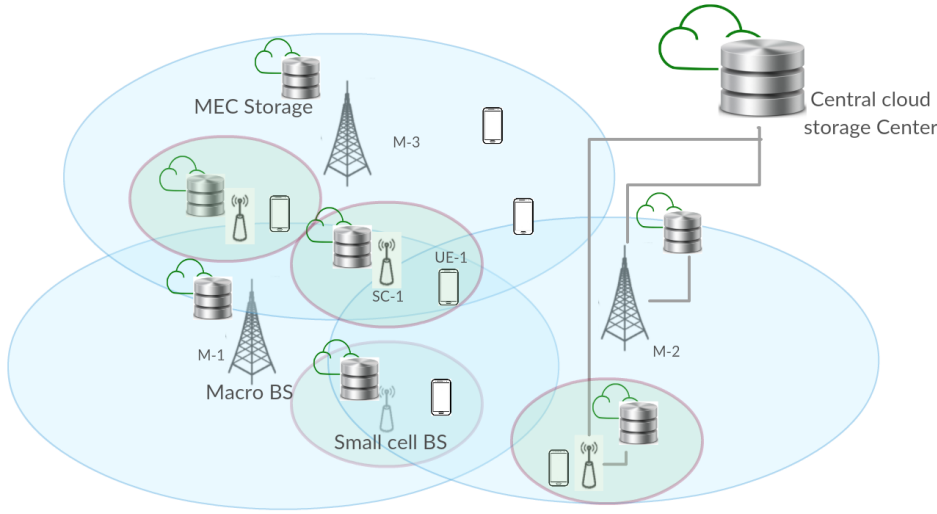


Figure 5.1: System Model.

In this setting, due to the overlapping coverage of the dense 5G HetNet, each user has multiple transmission opportunities to transfer its big data file through the available multiple wireless links. The overall goal is to achieve a reduction in transmission delay and energy dissipated by a UE during transmission of a big data file over the considered network.

In this Chapter, the same system model inherited from Chapter 4 (depicted in Fig. 5.1) is considered, where the HetNet is composed of multiple femto and macro BSs with overlapping coverage areas, and a public cloud storage system. Some of the details on the system parameters and the considered cell range expansion technique [18] are also inherited from Subsection 2.1.4 of Chapter 2. In the proposed system model, it is assumed that the UE can be associated to multiple BSs, where each BS association decision with a UE depends on the SINR experienced by the user on the downlink transmission link. In addition, it is also assumed that multiple paths between a UE and the cloud storage system may be available for data transfer purpose. Without loss of generality, it is assumed that the network condition does not change during a single storage execution.

This Chapter is organized as follows. First, the optimization problem is

presented. Second, the proposed solution in the form of an ACO based algorithm is described. Third, the simulation results of the proposed scheme are presented.

5.2 Optimization Problem

Our goal is to design a data transmission strategy which splits a file to be transmitted by the UE into a number of chunks and transmit them on the chosen data-links from the available multiple ones in the HetNet environment. Then, allocate the available PRBs to each of these data-links with the minimum possible transmission delay and consumed energy. The proposed method attempts to derive a data transmission strategy $(\rho_{i,j})$ and PRB allocation $(\zeta_{i,j})$ which minimizes the cost function in Equation (5.1).

$$(\rho_{i,j}, \zeta_{i,j}) = \arg \min_{\rho_{i,j}, \zeta_{i,j}} \{f(\rho_{i,j}, \zeta_{i,j})\} \quad (5.1)$$

$$f(\rho_{i,j}, \zeta_{i,j}) = \sum_{j=1}^{N_{DL}} \sum_{i=1}^{N_{UE}} \{e_{i,j}\} + \frac{\sum_{i=1}^{N_{UE}} \min_{j=1}^{N_{DL}} \left\{ \frac{D_{i,j}}{\beta_{i,j}} \right\}}{N_{UE}} \quad (5.2)$$

$$e_{i,j}(\rho_{i,j}, \zeta_{i,j}) = \frac{D_{i,j}}{\beta_{i,j}} \cdot \min \{P_{max}, (P_o \cdot \epsilon \cdot |\zeta_{i,j}|)\} \quad (5.3)$$

$$\beta_{i,j}(\rho_{i,j}, \zeta_{i,j}) = W_{i,j} \cdot \log_2 \left(\frac{\hat{P}_i^J}{\tau_{i,j} \cdot [1 + (\eta \div 100)]} \right) \quad (5.4)$$

$$\tau_{i,j} = \left(\sum_{k=1, k \in \cup_{i=1}^{N_{UE}^{total}} \{\zeta_i\}, k \neq j}^{N_{BS}^{total}} \hat{P}_i^j \right) \quad (5.5)$$

$$W_{i,j} = |\zeta_{i,j}| \cdot \Upsilon \quad (5.6)$$

The first term of (5.2) corresponds to the total energy consumed in transmitting a file from the UE to the cloud storage center while the second term corresponds to the total time delay incurred in the same transmission. The energy consumed by a data-link is calculated as per Equation(5.3). The derivation of this equation

is provided in the Appendix.

The description of all symbols and notations are given in Table 5.1

The maximum possible data rate on a wireless link is calculated as per Equation (5.4) following the Shannon-Hartley theorem [35]. The total interference experienced by the wireless link ($\tau_{i,j}$) is the sum of the received power from all BSs who shares at least one PRB allocated to the wireless link between the i^{th} UE and j^{th} BS as per Equation (5.5). The amount of bandwidth allocated to a wireless link ($W_{i,j}$) is the number of PRBs allocated to it multiplied by the bandwidth per PRB (Υ).

Table 5.1: NOTATIONS

Notation	Definition
ϕ, Q	Error correction syndromes
D_i	i^{th} bit of a data chunk
ψ_i	i^{th} bit of a padded data chunk
P	transmit power
\hat{P}	received power
GF	Galois field generator
$\rho_{i,j}$	Size of the data chunk assigned to data-link between i^{th} UE and j^{th} BS
$\zeta_{i,j}$	A set of PRBs assigned to the data-link between i^{th} UE and j^{th} BS
$\tau_{i,j}$	Total interference received by the data-link between i^{th} UE and j^{th} BS
$\beta_{i,j}$	Maximum possible data rate on the data-link between i^{th} UE and j^{th} BS
$W_{i,j}$	Amount of bandwidth assigned to the data-link between i^{th} UE and j^{th} BS
N_{UE}	Number of UEs in the network
N_{DL}^i	Number of data-links available to the i^{th} UE
$\epsilon_{i,j}$	Path loss of a wireless link between the i^{th} UE and j^{th} BS
α	Path loss compensation factor
\hat{P}_0	Target receive power
N_{ants}	Number of ants
N_{iter}	Number of maximum iterations
$v_{i,j}$	Number of PRBs assigned to the wireless link between i^{th} UE and j^{th} BS
Λ_k^{best}	Best solution of the k^{th} iteration
Λ_{all}^{best}	Best solution until the last iteration

5.3 Proposed ACO based Approach

The proposed ACO based scheme is shown in Fig. 5.2, which includes a preprocessing algorithm, an ACO based algorithm to discover the best possible data transmission strategy (i.e. data-links to be used by all the UEs), the PRB allocation to these data-links, and an error correction technique. Further details of this ACO based algorithm is described in Fig. 5.4.

$$N_{chunks}^i = N_{datapaths}^i - 2 \quad (5.7)$$

where $N_{datapaths}^i$ is the number of data paths allocated to the i^{th} UE by the ACO algorithm's data strategy.

In Fig. 3, the preprocessing step works as follows: Initially, all UEs begin by utilizing all the available data-links and splitting their data files. The size of the data chunk which would be transmitted over a data-link is proportional to the maximum possible data rate over the corresponding data-link. All the available PRBs are reused among all active UEs i.e. all the N_{PRB} PRBs are assigned to each UE. These N_{PRB} PRBs are equally assigned to all the available data links. This forms the initial solution. The proposed algorithm modifies this initial solution to generate the best possible solution.

5.3.1 Ant Colony Optimization Paradigm

ACO is a form of swarm intelligence [56] a system's property whereby the collective behaviours of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge [57]. ACO has been inspired from the functioning of ant colonies in the nature. Initially, the ants move randomly in the environment, upon finding a food they return to their colony while laying down pheromone trails. If an ant encounters such a pheromone trail it follows the path of trail instead of random movement. Over the time, pheromone

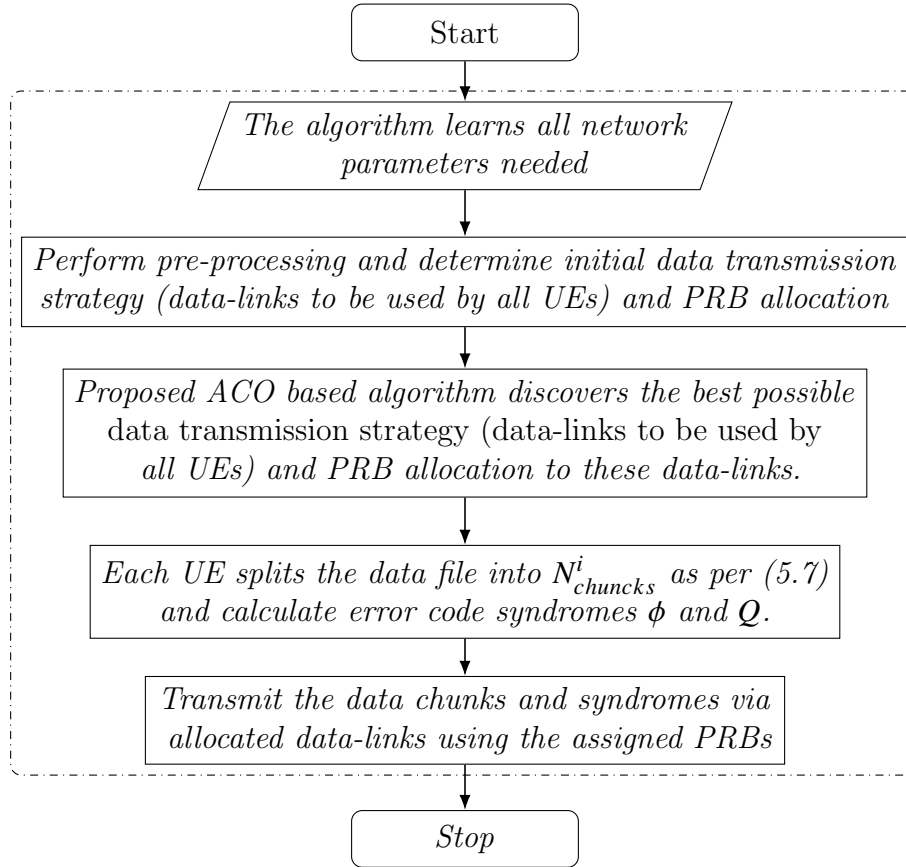


Figure 5.2: Proposed ACO-based solution made of a data transmission strategy and a PRB allocation and a loss data recovery scheme.

begins to evaporate and reduces its probability that it would be followed by the ants. On the longer path, it takes a longer time for an ant to travel from food source to nest and back, and this leads to more evaporation of pheromone. On the other hand, shorter paths encounter more trips by ants and experience denser deposit of pheromone on them, which makes them more attractive to be chosen as a path to be followed by fellow ants. Eventually, most of the ants end up traveling on the shortest path between the food source and the nest. Inspired by the foraging behaviour of ant colonies, in the ACO approach, a number of artificial ants build solutions to the considered combinatorial optimization problem and disclose the information on the quality of these solution to the ants in the next

cycle of the process.

5.3.2 Ant Colony Optimization Metaheuristic

Ant colony algorithm has been formalized into a meta-heuristic for solving combinatorial optimization problems [56]. In ACO, an artificial ant builds a solution by traveling across a graph $G(V,E)$, where V is the set of vertices and E is set of edges. This graph is constructed from a set of solution components by representing active UEs by the vertices and an ant constructs a solution incrementally by moving from one vertex to another along the edge. The ACO meta-heuristics is presented in Algorithm 3 [56].

Algorithm 3 The Ant Colony Optimization Metaheuristics

```

Set parameters, initialize pheromone trails
while Termination condition is not met do
    Construct Ant solutions
    apply Local search
    Update Pheromone
end while

```

The graph representation of our problem is depicted in Fig. 5.3. This representation prevails only after the preprocessing phase has been over. In this graph, the vertices represent the UEs in the network and during preprocessing, each UE is assigned a set of PRBs. An artificial ant randomly chooses a vertex to begin its journey from. Whenever an ant is on a vertex, the corresponding UE drops (chooses not to transmit the data chunk on) some of its data-links $DDL_i \in \{0, 1, 2, \dots, N_{DL}\}$. An ant travels through a complete graph with the conditions that it has to visit each vertex and each vertex can be visited only once. Whenever an UE decides to drop/to not use a data-link for transmission, the corresponding PRBs which were originally assigned to these dropped data-links would be available for reassignment to the rest of the active data-links. The updated transmission strategy along with the reassignment of all freshly available PRBs forms a possible candidate solution to the described problem.

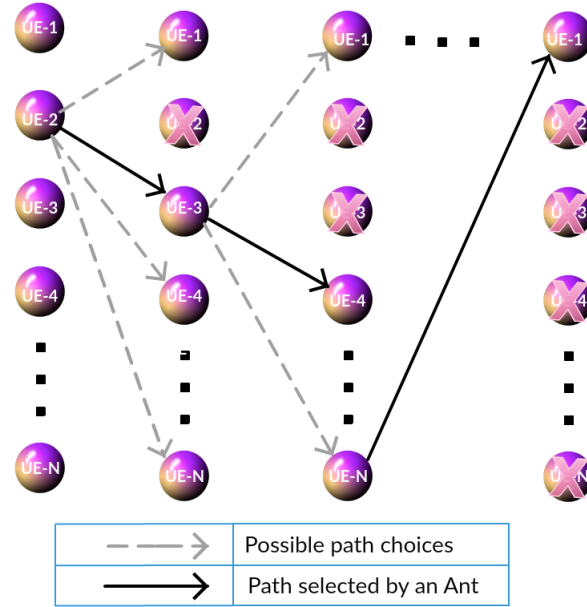


Figure 5.3: Travel map followed by an ant.

5.3.3 Proposed ACO Based Algorithm

The proposed ACO based algorithm for data transmission strategy and PRB allocation is described in Fig. 5.4. It is an evolutionary algorithm consisting of N_{ant} number of ants. During an iteration, these ants generates N_{ant} solutions. At the end of each iteration, the cost of each solution is calculated by means of the cost function given in Equation (5.2). The solution with the highest cost value is then selected as the best solution for that particular iteration. This best solution is used to update the information matrix which is passed to the subsequent iteration of the algorithm. This process is repeated for the number of iterations N_{iter} . The algorithm remembers the best solution among all the solutions of all the iterations ran so far. The termination condition for the proposed ACO algorithm is as follows:

$$(\text{Iteration index} \geq N_{iter}) \parallel (\sigma(f(\Lambda_{k-10}^{best}) \dots f(\Lambda_k^{best})) \leq \sigma_{min}) \quad (5.8)$$

where σ represents the standard deviation and σ_{min} is a predefined threshold value of the standard deviation.

At end of each iteration, the algorithm checks for the condition given in Equation (5.8); and if that is satisfied, the algorithm terminates and the best solution till the last iteration is passed to the main solution process. If the iteration index reaches the maximum number of iterations, then the algorithm terminates. The second condition is that if the last 10 iterations do not exhibit significant improvement in the cost of solution the algorithm terminates. The second condition is that if the last 10 iterations of the algorithm do not exhibit a significant improvement in the cost of the obtained solution, then the algorithm terminates. The satisfaction of this condition is considered as the convergence of the proposed iterative algorithm and the process is terminated to save the computational resources.

At each stage, an ant has to make two decisions before proceeding to the next UE/vertex. First, it has to choose which data-links to drop; and second, it has to choose which UE/vertex to move in next. In order to make a decision regarding which data-links to drop, the algorithm calculates the weights of each of the data-links as per Equation (5.9), and the data-links with the weight lower than a pre-determined threshold value is then selected and dropped from the transmission scheme.

$$WDL_k^i = \frac{D_k^i \cdot \rho_k}{\sum_{l=1}^{NDL^i} D_l^i \cdot \rho_l}, \quad (5.9)$$

where WDL_k^i is the weight of the k^{th} data link of the i^{th} UE, ρ_k is the total number of UEs covered by the BS which is an element of k^{th} data-link, D_k^i is the distance between the BS of the k^{th} data-link and i^{th} UE. The denominator of Equation (5.9) represents the weight of rest of the data-links available to the i^{th} UE, except the k^{th} data-link. l is the index used to represent the data-links available to the

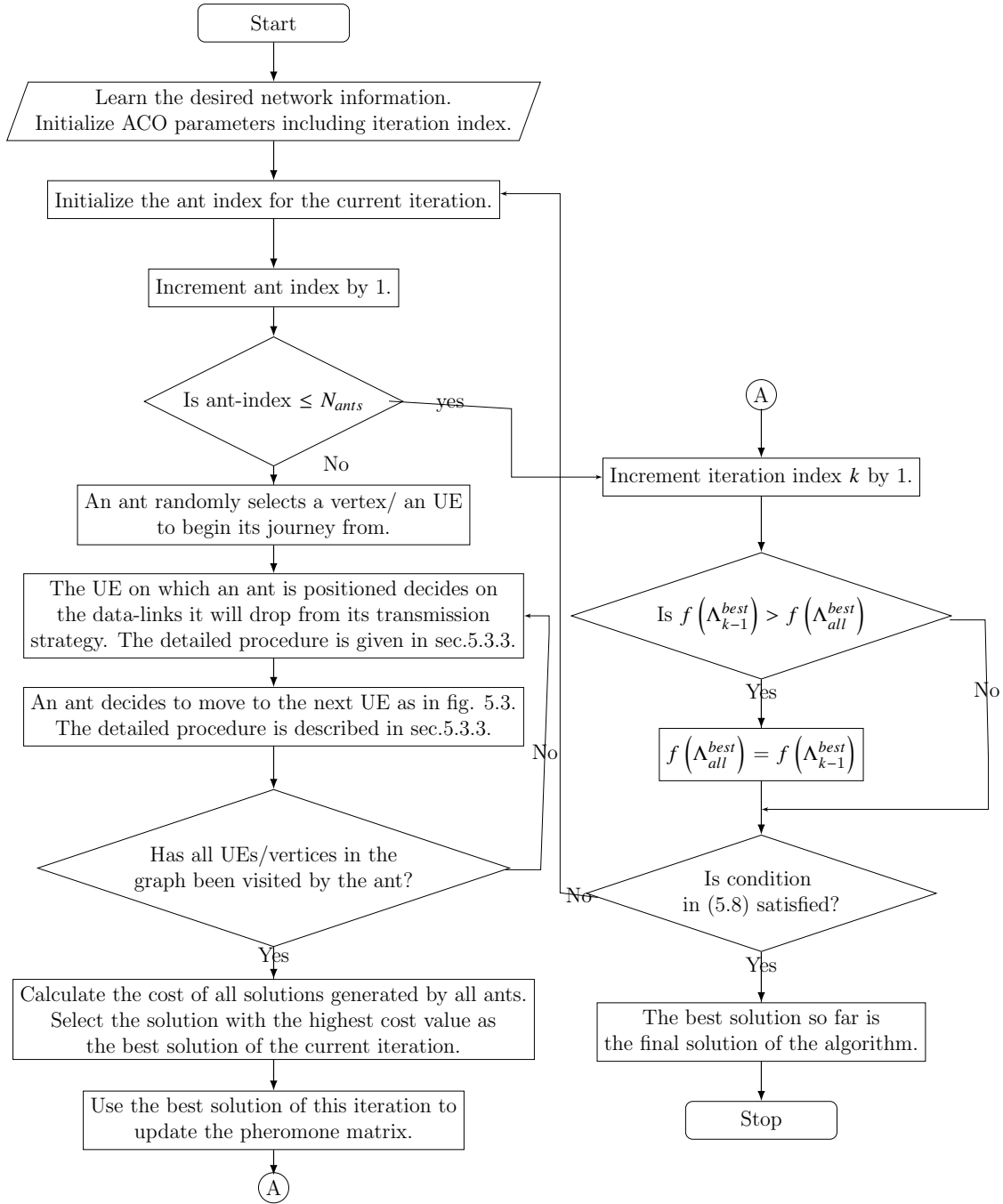


Figure 5.4: ACO based algorithm for data transmission strategy and PRB allocation

i^{th} UE $l = 1, \dots, NDL^i - 1$, where NDL^i is number of data links available to i^{th} UE.

Regarding the second decision, an ant chooses the UE/vertex to move next to, which is the one at minimum physical distance from the current UE compared to the rest of the UEs which have not been visited yet in the current journey of that particular ant.

5.3.4 Pheromone Definition

In the proposed algorithm, based on the principles of ACO, an information matrix is passed from an iteration to the subsequent iterations. The algorithm maintains a matrix related to the threshold value (γ) used by a UE to drop the data-links. An example of information matrix is shown in Table 5.2. The matrix consists of two columns. The first one consists of different quantified values of γ where $\gamma \in [0, 1]$ and the second one contains the weight of the corresponding γ values (W_γ).

Table 5.2: Information Matrix

Value of γ	Weight of γ (W_γ)
0.1	0.05
0.2	0.21
0.3	0.25
0.4	0.38
0.5	0.20
0.6	0.25
0.7	0.17
0.8	0.07
0.9	0.05
1	0.01

During the first iteration of the algorithm, the weights of all γ values are zero and all ants randomly select γ values in the range $[0, 1]$. At the end of each iteration, the solution which has the highest value of the cost function given in Equation (5.1) is selected as the best solution of the iteration. The weight (W_γ)

of the γ value used by the best solution is incremented by 0.01. In subsequent iterations, 20% of the ants randomly choose a γ value in the range $[0, 1]$ while the remaining 80% of ants use $\gamma = \hat{\gamma} - \iota$, where $\hat{\gamma}$ is the γ value corresponding to $\max(W_\gamma)$ and $\iota = \nu(-0.1, 0.1)$ where $\nu(-0.1, 0.1)$ is a uniform distribution between $[-0.1, 0.1]$. These random variables are added to avoid the optimization process getting stuck in a local minimum.

5.3.5 Lost Data Chunk Recovery Technique

In our proposed method, the lost data recovery technique implemented at the individual UE level (introduced in Chapter 3) is adopted. Also, the size of the data chunk is proportional to the up-link capacity of the data link on which it is to be transmitted. Each UE out of its total available data-links reserves two data links for the transmission of the data recovery syndromes ϕ and Q . It is considered that the size of ϕ (resp. Q) is the size of the largest data chunk. The smaller data chunks are padded with zeros as per Equation (5.11). These syndromes are calculated in a similar manner as was done in the design of the RAID-based algorithm for faster data transfer: Case of Data Chunk Size Proportional to Up-link Capacity scheme (introduced in Chapter 3), i.e. using Equations (5.12)-(5.13) and are saved to the cloud system along with the data chunks of the file.

$$Z = \max \text{size}(D_i) \quad (5.10)$$

$$\psi_i(k) = \begin{cases} D_i(k) & \text{for } k \leq \text{size}(D_i) \\ 0 & \text{for } \text{size}(D_i) < k \leq Z \end{cases} \quad (5.11)$$

$$\phi_1 = \psi_1 \quad \text{where, } D_i \text{ is } i^{th} \text{ data chunk.}$$

$$\phi_i = \phi_{i-1} \oplus \psi_i, \quad \text{for } 2 \leq i \leq N_i^{chunks}$$

$$\phi = \{\phi_1, \dots, \phi_{N_i^{chunks}}\} \quad (5.12)$$

$$Q_1 = GF^1 \cdot \psi_1$$

$$\begin{aligned}
Q_i &= Q_{i-1} \oplus GF^1 \cdot \psi_i \\
Q &= \{Q_1, \dots, Q_{N_i^{chunks}}\}
\end{aligned} \tag{5.13}$$

5.4 Performance Evaluation of the Proposed ACO-Based Scheme

The proposed algorithm is executed on the simulated network model executed on MATLAB (see Section 2.1.4). A typical simulated network scenario is shown in Fig. 5.5, where the HetNet consists of two macrocells namely M1 and M2 and 3 femtocells F1, F2, F3. The coverage areas of the macrocells are denoted by blue circles while the coverage areas of the femto cells are denoted by green circles. A number of UEs denoted by red diamonds are randomly placed all over the network. For comparison purpose, the same Greedy and Data Split Multiple UE (DSMU) algorithms introduced in Chapter 4 are considered and executed on the same network model. The performance of our proposed algorithm is compared

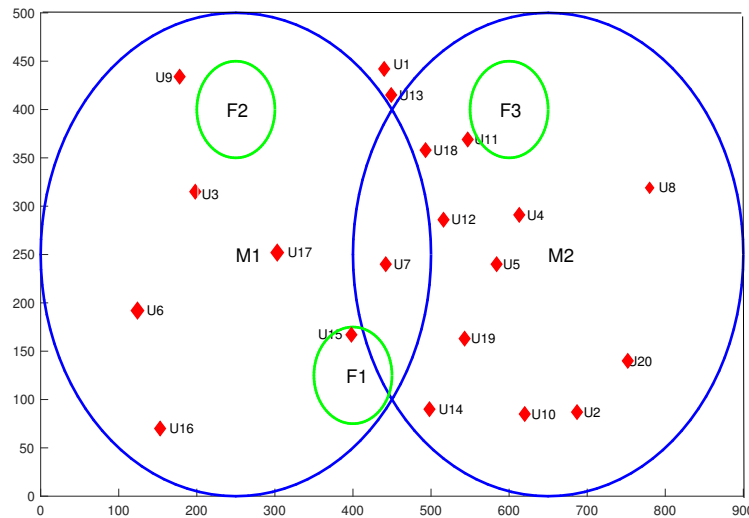


Figure 5.5: Simulated instance of network scenario.

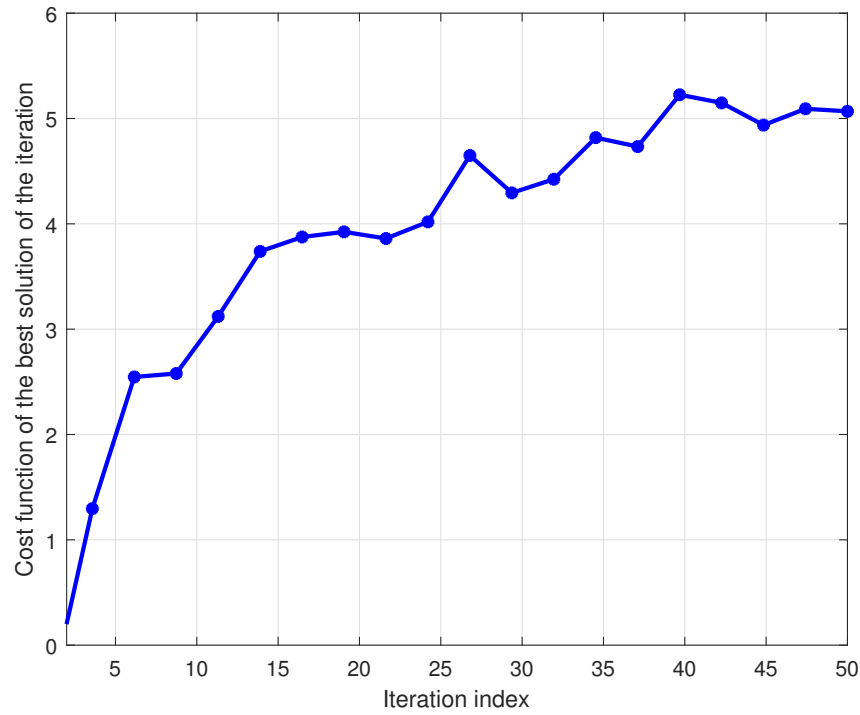


Figure 5.6: Convergence of the proposed algorithm.

against that of those algorithms.

5.4.1 Convergence of the Proposed ACO Algorithm

The proposed ACO based algorithm is an evolutionary algorithm whose design is inspired by the principles of the biological evolution. The convergence of the proposed algorithm is illustrated in Fig. 5.6. The value of the the cost function (in Equation (5.2)) at the end of cycle is plotted against the number of cycles that the algorithm has been executed for. It is observed that the algorithm converges after 30 cycles. Therefore, for the generation of the remaining graphs, the proposed algorithm has been executed for 35 cycles.

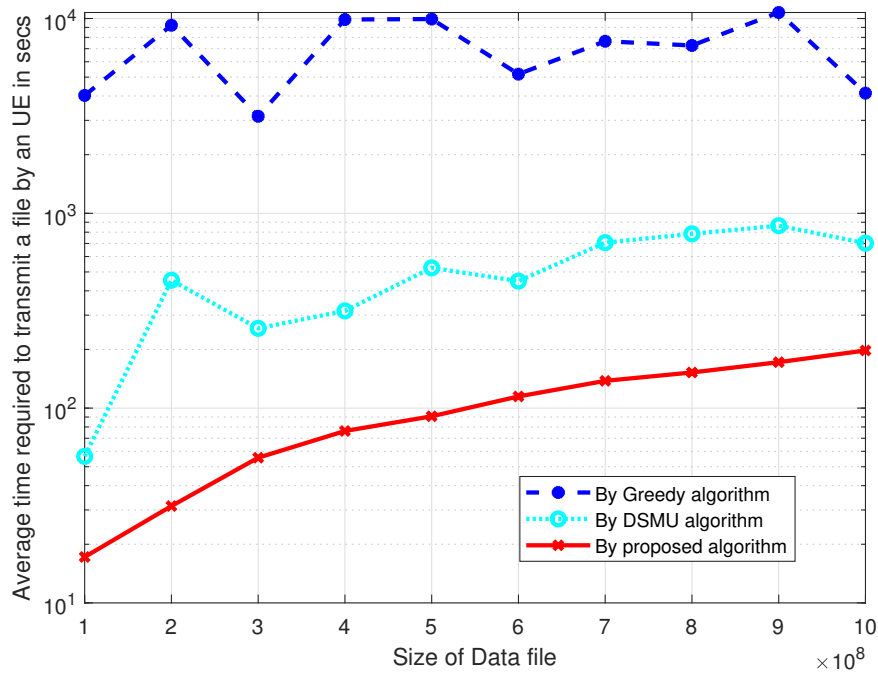


Figure 5.7: Impact of data transmission delay on the data file size

5.4.2 Simulation Results

For our simulations, all active UEs in the network are assumed to have a data file of the same size to transmit to the cloud storage system.

First, the data file size is varied from 100 Mbits to 1 Gbits and the impact of this variation on the average delay experienced by all UEs is plotted in Fig.5.7. From this figure, it is observed that the time required by our algorithm for complete file transfer is significantly lower compared to that needed by the greedy and DSMU algorithms. Indeed, to transfer a data file of size 900 Mbits, the greedy algorithm needed $1 \cdot 10^4$ secs and DSMU algorithms required approximately $9 \cdot 10^2$ secs while our proposed algorithm transfers the file in $2 \cdot 10^2$ secs, which is 91% reduction in transmission delay compared to greedy algorithm and 77.78% reduction in transmission delay compared to DSMU algorithm.

Second, the number of active UEs in the network is varied and the impact of

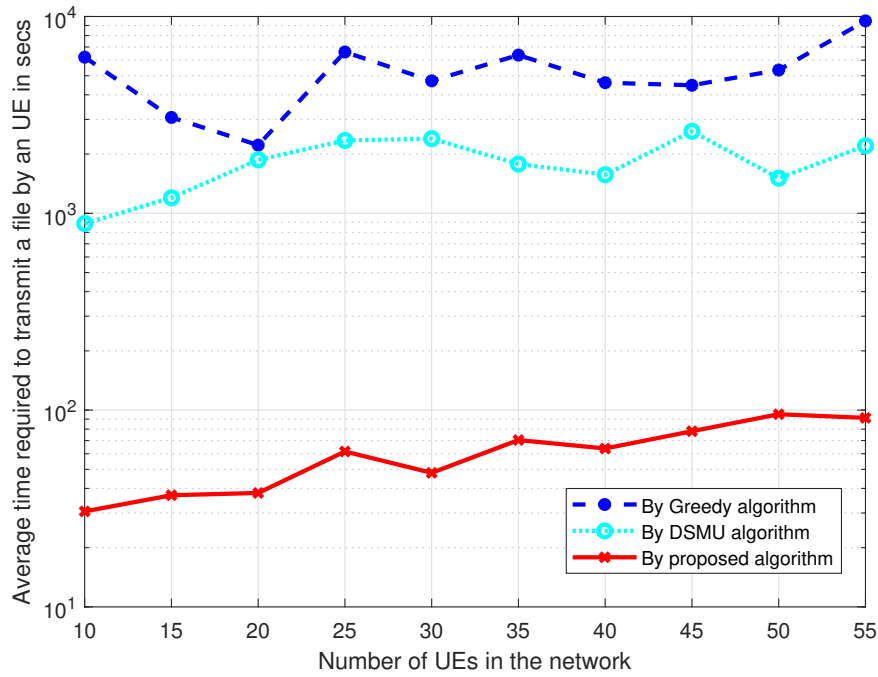


Figure 5.8: Data transmission delay when the number of UEs in the network is increased

this variation on the average delay experienced by all UEs is plotted in Fig. 5.8. It can be observed that for the proposed scheme, the average delay experienced by a UE is approximately 50 times less than that experienced by the greedy algorithm. Also, the delay experienced by the DSMU algorithm is 20 times higher than the one experienced by the proposed algorithm.

Third, the number of femto cell BSs in the network is increased and the impact of this increase on the average delay experienced by all UEs is plotted in Fig. 5.9. It can be observed that the data transmission delay generated using our proposed algorithm is about 10^2 times lower than that generated by the greedy and 8 times lower than that generated by the DSMU algorithm. Moreover, it can be noted that for the DSMU and the proposed algorithm, as the number of small cell BSs increases in the network, the delay encountered during transmission of a data file is reduced significantly. This is due to the fact that both of

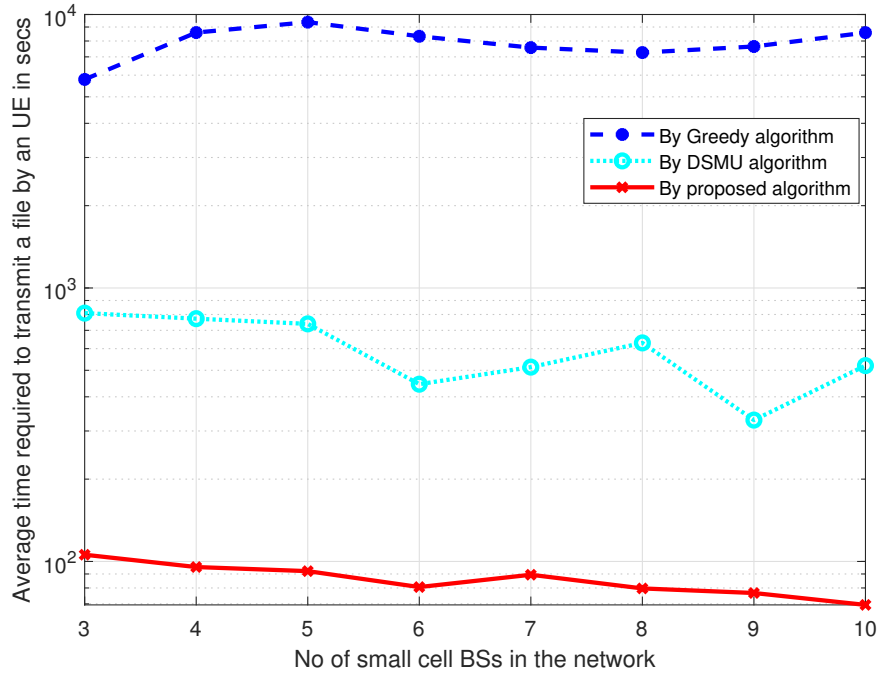


Figure 5.9: Impact of data transmission delay when the number of small cell BSs in the network is increased

these algorithms splits the data file into multiple chunks and transmit them simultaneously over multiple available data-links. As the number of BSs available in the network increases, the number of data-links available to a UE in the network increases, which justifies this reduction in the delay. This graph proves that the enhanced infrastructure of 5G HetNet is efficiently used by our proposed ACO based algorithm.

Besides reducing the data transmission delay in the considered network, the proposed algorithm also focuses on minimizing the energy consumption in the network. The energy consumed by a UE is calculated as per Equation (5.14), i.e.

$$e_i = \sum_{j \in J} e_{ij} \quad (5.14)$$

$$e_{ij} = \min \{P_{max}, (P_o \cdot \epsilon \cdot v_{i,j})\} \cdot \frac{D_{ij}}{\beta_{ij}} \quad (5.15)$$

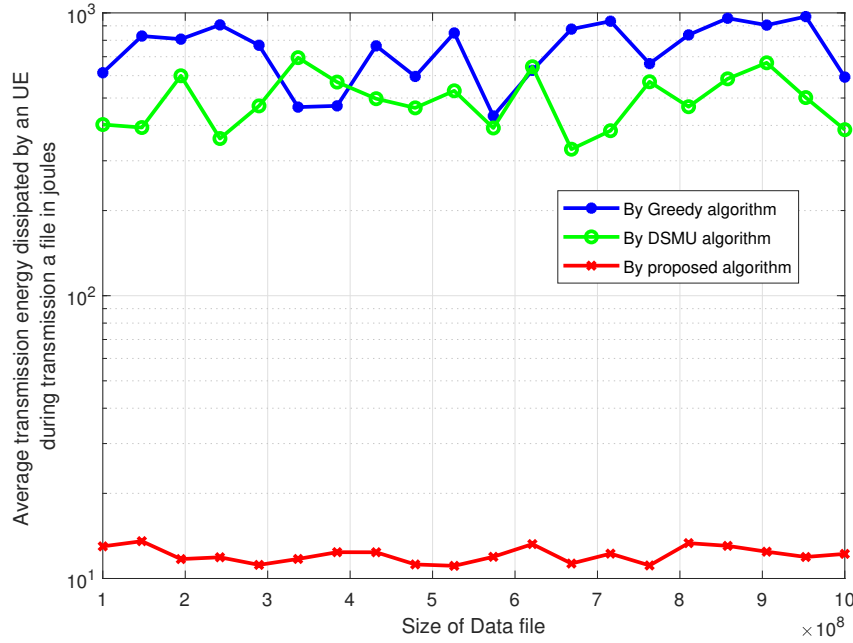


Figure 5.10: Energy dissipated by a UE during transmission when the data file size is increased

where, e_i is the amount of energy transmitted by the i^{th} UE, J is a set of PRBs allocated to the i^{th} UE and e_{ij} is the amount of energy transmitted by j^{th} data-link of i^{th} data-link. The derivation of Equation (5.15) can be found in Appendix 5.5.1.

Fourth, Now, let's assume that all active UEs in the network have a data file of equal size to store in the cloud storage system. The data file size is varied from 100 Mbits to 1 Gbits and the impact of this variation on the average delay experienced by all UEs is investigated. The results are captured in Fig. 5.10, where it can be observed that the energy transmitted by our proposed algorithm for complete file transfer is significantly lower compared to the amount of energy transmitted by the greedy and DSMU algorithms. This improvement in performance is attributed to the fact the energy consumed by the network is Part of the objective function and is among the constraints of the optimization problem described in Equations (5.1-5.6). The optimal solution also aimed at minimizing

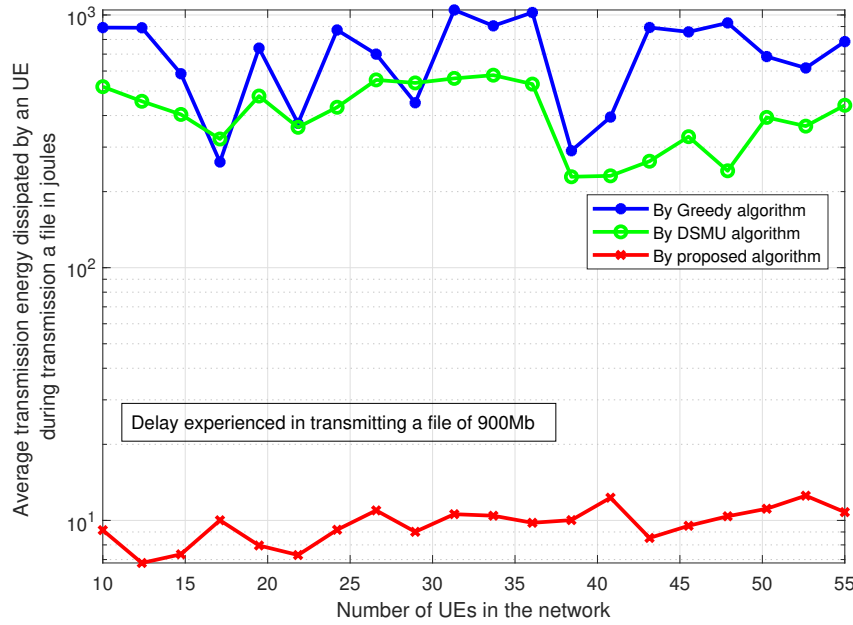


Figure 5.11: Energy dissipated by a UE during transmission of the data file when the number of UEs in the network is increased

this energy consumption. Indeed, for the DSMU and Greedy algorithms, an average energy of about 5×10^2 Joules was consumed to transfer a data file of size 900 Mbits whereas for our proposed algorithm, about 105 Joules was consumed for the same.

Fifth, the number of active UEs in the network is varied and the impact of this variation on the average delay experienced by all UEs is plotted in Fig. 5.11. It is observed that the average energy transmitted by our proposed scheme is approximately 10^2 times less than that transmitted by the greedy algorithm or DSMU algorithm. It can be further observed that the performance of proposed algorithm is less sensitive to changes in the number of active UEs while the performance of the greedy algorithm fluctuates with the number and locations of the active UEs in the network. The rationale behind this stable performance of the proposed algorithm with the variations in the number and locations of the UEs is that the proposed algorithm considers energy consumption while deciding on the

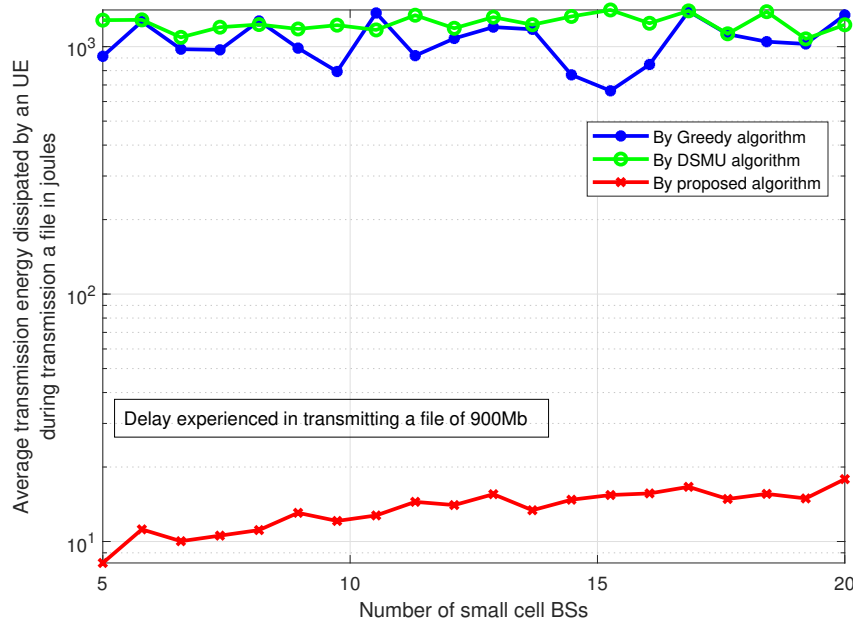


Figure 5.12: Performance of proposed and other comparison algorithms to varying number of small cell BSs in the network.

data transmission strategy and PRB allocation algorithm considers the network conditions, including the locations and number of active UEs in the network. On the other hand, the greedy and DSMU algorithms do not take into account the factor of minimizing the energy consumption.

Sixth, the number of small cell BSs in the network is increased and the impact of this increase on the average energy transmitted by all UEs is investigated. The results are shown in Fig. 5.12. It can be observed that the average energy transmitted by a UE using our proposed algorithm is lower than that generated by the greedy and DSMU algorithms. Moreover, compared to the DSMU and greedy algorithms, our proposed algorithm is less sensitive to the variation in number of small BSs in the network.

5.5 Appendix for Chapter.5

5.5.1 Proof of energy consumption equation

As per the physics law, power is the energy per unit time. Therefore, the amount of energy dissipated is formulated as

$$e = P \cdot t, \quad (5.16)$$

where e is the energy dissipated, P is the power transmitted and t is the amount of time the power P is transmitted for. The LTE UE's transmission power P in a subframe of the Physical Uplink Shared Channel (PUSCH) [58] onto which the UE data is transmitted is obtained as:

$$P = \min \{P_{max}, (P_o + \varsigma \cdot \epsilon + 10 \cdot \log_{10} M + \Delta_{TF} + f)\} \quad \text{dBm} \quad (5.17)$$

where P is the uplink transmit power, P_{max} is the maximum transmission power, which is 23 dBm ± 2 dB for a class 3 UE, P_o is a power offset, M is the number of PRBs assigned to the UE, ς is the path loss compensation factor where $\varsigma = 1$ corresponds to the conventional open loop power control while $0 < \varsigma < 1$ corresponds to the fractional open loop power control scheme, Δ_{TF} and f are UE specific parameters which are based on the modulation and coding scheme.

For simplicity purposes, we assume that the parameters Δ_{TF} , f and ς are constant for all UEs; therefore, they can be assumed to be negligible; hence Equation (5.17) becomes

$$P = \min \{P_{max}, (P_o \cdot \epsilon \cdot M)\} \quad (5.18)$$

Therefore, the energy dissipated in the transmission of a data file/chunk of

size D_{ij} via the data-link between the i^{th} UE and the j^{th} data center is

$$e_{ij} = \min \{P_{max}, (P_o \cdot \epsilon \cdot \nu_{i,j})\} \cdot \frac{D_{ij}}{\beta_{ij}}$$

Chapter 6

Conclusion

6.1 Conclusion

WE have considered a composite wireless network architecture integrating HetNet, MCC and MEC technologies. The goal of thesis is to efficiently transmit a massive data files to cloud data centers. Different solutions following divergent approaches have proposed.

The first method attempts to achieve the goal through efficiently utilizing the available infrastructure resources of HetNet, MCC and MEC networks. The proposed technique exploits the fact that in the densely deployed HetNet, most of the UEs receives an overlapping coverage from multiple BSs. In order to expedite the data transmission process, a big data to be transmitted is split into multiple data chunks, which are concurrently transmitted over multiple available data paths. A RAID based lost data recovery technique is embedded in the proposed algorithm to deal with the high probability of failure to retrieve a stored data chunks. The simulated results show that the proposed method outperforms the conventional data storage method between the mobile device and the cloud system. The proposed method was further improved in another proposed algorithm wherein the big data file to be transmitted to cloud data is segmented into chunks

where the size of each chunk is directly proportional to the data rate capacity of the data-link on which it is supposed to be transmitted. Another variation of the data transmission technique is specifically designed for an efficient transmission of big video files wherein, a big data file is compressed using the HEVC compression algorithm. Then, the compressed file is segmented and stored to the cloud data center along with the lost data recovery syndromes.

In all the above proposed approaches, the massive data file storage task is treated as a standalone activity by a single UE. In the subsequent solution approaches, we have considered the same problem as a centralized one from a network resource management perspective. The radio and infrastructure management problem is formulated as a combinatorial optimization problem, and two different solutions based on different approaches have been proposed.

First, a game theoretic mechanism design approach has been proposed to solve the problem. Here, the infrastructure and radio resource management problem in the considered network integrating HetNet, MCC and MEC technologies is formulated as an optimization problem. Since all UEs in the network compete for limited radio resources, the radio resources allocation problem is formulated as a non-cooperative game and graph colouring problem. The proposed algorithmic solution framework uses a mechanism design technique, and it consists of a storage intensive task offloading scheme, a PRB allocation scheme and a pricing algorithm. The performance of the proposed solution framework is compared against that of the conventional DSMU algorithm and the proposed solution is shown to outperform the DSMU algorithm. Moreover, through mathematical analysis, the proposed algorithm has proven to achieve several desirable properties such as incentive compatibility, individual rationality and efficiency.

Second, another solution is presented in the form of an ant colony optimization to solve the task offloading and radio resource management problem. The considered problem is formulated as a multi-objective optimization problem, with the goal to minimize the data transmission delay and energy consumption during

data transmission. The solution algorithm is proposed based on the ant colony optimization technique. The performance of the proposed algorithm is compared against that of the Greedy and DSMU algorithms on THE simulated network model, showing its superiority.

6.2 Future Work

As future work, one can investigate the design of a revenue sharing scheme to complement the proposed solution frameworks (mechanism design based and ACO based approaches), in which multiple MNOs operating various macro and small cell BSs and CSPs will be sharing the revenue generated from their coalition with fairness so that the coalition will function successfully for a longer duration.

Another way to extend the presented work is to enhance the presented data offloading scheme specifically tailored for the type of data file to be transmitted such as text, audio, image, to a name a few.

References

- [1] R. Minerva, A. Biru, and D. Rotondi, “Towards a definition of the internet of things (iot),” *IEEE Internet Initiative*, vol. 1, pp. 1–86, 2015.
- [2] Y. Liu, J. E. Fieldsend, and G. Min, “A framework of fog computing: Architecture, challenges, and optimization,” *IEEE Access*, vol. 5, pp. 25 445–25 454, 2017.
- [3] A. Van Kempen, T. Crivat, B. Trubert, D. Roy, and G. Pierre, “Mec-conpaas: An experimental single-board based mobile edge cloud,” in *Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, 2017 5th IEEE International Conference on. IEEE, 2017, pp. 17–24.
- [4] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks,” *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45–55, 2014.
- [5] H. S. C. Glaucio, W. Isaac, A. Alagan, and J. Muhammad, “Analysis of joint parallelism in wireless and cloud domains on mobile edge computing over 5G systems,” *Journal of Communications and Networks*, vol. 20, no. 6, pp. 1–12, 2018.
- [6] 3GPP, “Improvement of radio resource management (rrm) across rns and rns/bss,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 3GPP TR 25.881, 12 2018, version 5.0.0. [On-

- line]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1314>
- [7] L. Gkatzikis and I. Koutsopoulos, “Migrate or not? exploiting dynamic task migration in mobile cloud computing systems.” *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 1–0, 2013.
 - [8] G. H. S. Carvalho, I. Woungang, A. Anpalagan, M. Jaseemuddin, and E. Hosain, “Intercloud and hetnet for mobile cloud computing in 5G systems: Design issues, challenges, and optimization,” *IEEE Network*, vol. 31, no. 3, pp. 80–89, 2017.
 - [9] X. Chen, “Decentralized computation offloading game for mobile cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.
 - [10] H. Liang, L. X. Cai, D. Huang, X. Shen, and D. Peng, “An smdp-based service model for interdomain resource allocation in mobile cloud networks,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 5, pp. 2222–2232, 2012.
 - [11] R. Siddavaatam, I. Woungang, G. Carvalho, and A. Anpalagan, “Efficient ubiquitous big data storage strategy for mobile cloud computing over hetnet,” in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.
 - [12] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, “Raid: High-performance, reliable secondary storage,” *ACM Computing Surveys (CSUR)*, vol. 26, no. 2, pp. 145–185, 1994.
 - [13] R. Siddavaatam, I. Woungang, G. Carvalho, and A. Anpalagan, “An efficient method for mobile big data transfer over hetnet in emerging 5G systems,”

- in *Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, 2016 IEEE 21st International Workshop on. IEEE, 2016, pp. 59–64.
- [14] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand *et al.*, “Overview of the high efficiency video coding(HEVC) standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
 - [15] R. Siddavaatam, I. Woungang, H. S. C. Glaucio, and A. Anpalagan, “Mobile cloud storage over 5G: A mechanism design approach,” *IEEE Systems Journal*, vol. Manuscript ID ISJ-RE-18-06770, (Accepted on June 8, 2018).
 - [16] R. K. Dash, A. Rogers, and N. R. Jennings, “A mechanism for multiple goods and interdependent valuations,” in *International Workshop on Agent-Mediated Electronic Commerce*. Springer, 2004, pp. 15–29.
 - [17] R. Siddavaatam, I. Woungang, and A. Anpalagan, “Joint optimization of radio and infrastructure resources for energy efficient massive data storage in mobile cloud over 5G hetnet,” *IET Wireless Sensor Systems*, (Ref: WSS-2019-0015), (Accepted on March 18, 2019).
 - [18] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, “Algorithms for enhanced inter-cell interference coordination (eicic) in LTE hetnets,” *IEEE/ACM transactions on networking*, vol. 22, no. 1, pp. 137–150, 2014.
 - [19] “Microsoft Azure data analytic tools,” <https://azure.microsoft.com/en-us/product-categories/analytics/>, accessed: 2019-01-13.
 - [20] “Data lakes and analytics on AWS,” <https://aws.amazon.com/big-data/datalakes-and-analytics/>, accessed: 2019-01-13.
 - [21] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. Silva, F. Tufveson, A. Benjebbour, and G. Wunder, “5G: A tutorial overview of standards,

- trials, challenges, deployment, and practice,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.
- [22] K. Zheng, F. Hu, W. Wang, W. Xiang, and M. Dohler, “Radio resource allocation in LTE-advanced cellular networks with M2M communications,” *IEEE communications Magazine*, vol. 50, no. 7, 2012.
 - [23] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, “M2M scheduling over LTE: Challenges and new perspectives,” *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 34–39, 2012.
 - [24] S.-Y. Lien, K.-C. Chen, and Y. Lin, “Toward ubiquitous massive accesses in 3GPP machine-to-machine communications,” *IEEE Communications Magazine*, vol. 49, no. 4, 2011.
 - [25] L. Saker, S.-E. Elayoubi, R. Combes, and T. Chahed, “Optimal control of wake up mechanisms of femtocells in heterogeneous networks,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 664–672, 2012.
 - [26] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, “Challenges on wireless heterogeneous networks for mobile cloud computing,” *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, 2013.
 - [27] C. Chang, S. N. Srirama, and R. Buyya, “Internet of things (IoT) and new computing paradigms,” *arXiv preprint arXiv:1812.00591*, 2018.
 - [28] X. Sun and N. Ansari, “EdgeIoT: Mobile edge computing for the internet of things,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.
 - [29] N. Grozev and R. Buyya, “Inter-cloud architectures and application brokering: taxonomy and survey,” *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014.

- [30] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. Da Silva, “Waveform and numerology to support 5G services and requirements,” *IEEE Communications Magazine*, vol. 54, no. 11, pp. 90–98, 2016.
- [31] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, “Algorithms for enhanced inter-cell interference coordination (eicic) in LTE hetnets,” *IEEE/ACM transactions on networking*, vol. 22, no. 1, pp. 137–150, 2014.
- [32] R. Siddavaatam, A. Anpalagan, I. Woungang, and S. Misra, “Ant colony optimization based sub-channel allocation algorithm for small cell hetnets,” *Wireless personal communications*, vol. 77, no. 1, pp. 411–432, 2014.
- [33] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.
- [34] J. Zyren and W. McCoy, “Overview of the 3GPP long term evolution physical layer,” <https://www.nxp.com/docs/en/white-paper/3GPPEVOLUTIONWP.pdf>, 2007.
- [35] C. E. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.
- [36] M. Felemban, S. Basalamah, and A. Ghafoor, “A distributed cloud architecture for mobile multimedia services,” *IEEE Network*, vol. 27, no. 5, pp. 20–27, 2013.
- [37] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, “A framework for cooperative resource management in mobile cloud computing,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, 2013.
- [38] Y. Choi, H. Kim, S.-w. Han, and Y. Han, “Joint resource allocation for parallel multi-radio access in heterogeneous wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3324–3329, 2010.

- [39] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *Infocom, 2012 Proceedings IEEE*. IEEE, 2012, pp. 945–953.
- [40] N. Zhang, N. Cheng, A. T. Gamage, K. Zhang, J. W. Mark, and X. Shen, “Cloud assisted hetnets toward 5G wireless networks,” *IEEE communications magazine*, vol. 53, no. 6, pp. 59–65, 2015.
- [41] M. Shiraz, M. Sookhak, A. Gani, and S. A. A. Shah, “A study on the critical analysis of computational offloading frameworks for mobile cloud computing,” *Journal of Network and Computer Applications*, vol. 47, pp. 47–60, 2015.
- [42] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, “Energy-efficient fault-tolerant data storage and processing in mobile cloud,” *IEEE Transactions on cloud computing*, vol. 3, no. 1, pp. 28–41, 2015.
- [43] M. Gerasimenko, D. Moltchanov, R. Florea, S. Andreev, Y. Koucheryavy, N. Himayat, S.-P. Yeh, and S. Talwar, “Cooperative radio resource management in heterogeneous cloud radio access networks,” *IEEE Access*, vol. 3, pp. 397–406, 2015.
- [44] J. Liu and Q. Zhang, “Offloading schemes in mobile edge computing for ultra-reliable low latency communications,” *IEEE Access*, vol. 6, pp. 12 825–12 837, 2018.
- [45] P. Zhao, H. Tian, C. Qin, and G. Nie, “Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing,” *IEEE Access*, vol. 5, pp. 11 255–11 268, 2017.
- [46] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, “Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks,” *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

- [47] H. Cao and J. Cai, “Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–764, 2018.
- [48] R. Siddavaatam, I. Woungang, and S. K. Dhurandher, “An efficient data transmission technique for big video files over hetnet in emerging 5G networks,” in *Advanced Information Networking and Applications (AINA-2019)*. IEEE, 2019 (Accepted Dec 17, 2018).
- [49] C. U. Castellanos, D. L. Villa, C. Rosa, K. I. Pedersen, F. D. Calabrese, P.-H. Michaelson, and J. Michel, “Performance of uplink fractional power control in utran LTE,” in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. IEEE, 2008, pp. 2517–2521.
- [50] H. P. Anvin, “The Mathematics of RAID-6,” <http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf>, 2007, [Online; Linux Kernel Reference].
- [51] V. Sze, M. Budagavi, and G. J. Sullivan, “High efficiency video coding (hevc), integrated circuits and systems,” *Cham: Springer*, 2014.
- [52] S.-Y. Lien, S.-L. Shieh, Y. Huang, B. Su, Y.-L. Hsu, and H.-Y. Wei, “5G new radio: Waveform, frame structure, multiple access, and initial access,” *IEEE communications magazine*, vol. 55, no. 6, pp. 64–71, 2017.
- [53] N. Nisan and A. Ronen, “Algorithmic mechanism design,” *Games and Economic behavior*, vol. 35, no. 1-2, pp. 166–196, 2001.
- [54] J. S. Plank, “A tutorial on reed–solomon coding for fault-tolerance in raid-like systems,” *Software: Practice and Experience*, vol. 27, no. 9, pp. 995–1012, 1997.
- [55] V. Krishna, *Auction theory*. Academic press, 2009, ISBN: 9780123745071.

- [56] M. Dorigo and T. Stützle, “Ant colony optimization: overview and recent advances,” *Techreport, IRIDIA, Universite Libre de Bruxelles*, vol. 8, 2009.
- [57] J. Fulcher, “Computational intelligence: an introduction,” in *Computational intelligence: a compendium*. Springer, 2008, pp. 3–78.
- [58] M. Lauridsen, A. R. Jensen, and P. E. Mogensen, “Reducing LTE uplink transmission energy by allocating resources.” in *VTC Fall*, 2011, pp. 1–5.

Glossary

5G network Fifth Generation cellular mobile communication network. 1, 6, 7

ACO Ant Colony Optimization. iii, 6, 9

AP Access Point. 2

BS Base Station. 2

CRE cell range expansion. 7

CRRM Common Radio Resource Manager. 3, 15

CRRMS Common Radio Resource Management Server. 17, 18

CSB Cloud Service Broker. 2, 6, 17

CSP Cloud Service Provider. 2, 6

DSMU Data Split Multiple UE. iii, 6, 8–10

H2H Human 2 Human communication. 13, 15

HetNet Heterogeneous Network. 1, 6, 13

HEVC High Efficiency Video Coding. 49, 53

IoT Internet of Things. 1, 11

- JRRM** joint radio resource management. 14
- LTE** Long Term Evolution. 19
- M2M** Machine 2 Machine communication. 12, 13, 15, 18
- MCC** Mobile Cloud Computing. iii, 1, 7
- MEC** Mobile Edge Computing. iii, 7, 16
- MNO** Mobile Network Operator. 1, 6, 16
- NR** New Radio. 19
- PRB** physical resource block. 6
- QoS** quality of service. 13, 14
- RAID** Redundant Array of Independent Disks. iii, 6
- RAT** radio access technology. 3
- SINR** signal to interference and noise ratio. 5, 7, 21, 35, 60
- UE** User Equipment. 3
- VCG** Vickrey-Clarke-Groves. 66, 70, 72
- VM** Virtual Machine. 17