ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

DOTTORATO DI RICERCA IN

Elettronica, Telecomunicazioni
e Tecnologie dell'Informazione

Ciclo XXVIII

Settore Concorsuale di afferenza: 09/F2

Settore Scientifico Disciplinare: ING-INF/03

RESOURCES OPTIMIZATION FOR DISTRIBUTED MOBILE PLATFORMS IN SMART CITIES

Presentata da: Daniela Mazza

Relatore:

Prof. Giovanni Emanuele Corazza

Coordinatore Dottorato:

Prof. Alessandro Vanelli Coralli

Correlatore:

Prof. Daniele Tarchi

Esame finale anno 2016

ALMA MATER STUDIORUM – UNIVERSITY OF BOLOGNA

RESOURCES OPTIMIZATION FOR DISTRIBUTED MOBILE PLATFORMS IN SMART CITIES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL, ELECTRONIC AND INFORMATION ENGINEERING (DEI) GUGLIELMO MARCONI
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Coordinator: Prof. Alessandro Vanelli Coralli
Supervisor: Prof. Giovanni Emanuele Corazza
Advisor: Prof. Daniele Tarchi

Daniela Mazza Candidate
March 2016

To my family

”I am struck by how, except when you’re young, you really need to prioritize in life, figuring out in what order you should divide up your time and energy. If you don’t get that sort of system set by a certain age, you’ll lack focus and your life will be out of balance.”

Haruki Murakami, *What I Talk About When I Talk About Running*

Abstract

This thesis collects the outcomes of a Ph.D. course in Electronics, Telecommunications, and Information Technologies Engineering and it is focused on the study and design of techniques able to optimize resources in distributed mobile platforms. It is related to a typical *smart city* environment, in order to enhance quality, performance and interactivity of urban services. The subject is the operation of *computation offloading*, intended as the delegation of certain computing tasks to an external platform, such as a cloud or a cluster of devices. Offloading the computation tasks can effectively expand the usability of mobile devices beyond their physical limits and may be necessary due to limitations of a system handling a particular task on its own.

The computation offloading within an ecosystem as a urban community, where a large amount of users are connected towards even multiple devices, is a challenging subject. In a very close future, smart cities will be peculiar sources of intensive computing tasks, since they are conceived as systems where e-governance will be not only transparent and fast, but also oriented to energy and water conservation, efficient waste disposal, city automation, seamless facilities to travel and affordable access to health management systems. Also traffic will need to be monitored intelligently, emergencies foreseen and resolved quickly, homes and citizens provided with a wide series of control and security devices. All these ambitious aspirations will require the deployment of infrastructures and systems where devices will generate massive data and should be orchestrated in a collective way, to pursue synergic goals. In this context, the computation offloading is an operation dealing with the optimization of urban services, in order to reduce costs and consumption of resources and to improve the connection between citizens and government.

This dissertation is organized in three main parts, dealing with the optimization of the resources in a smart city background from different points of view.

The first part introduces the Urban Mobile Cloud Computing (UMCC) framework, a system model that takes into account a series of features related to Heterogeneous Networks (HetNets), cloud architectures, various characteristics of the Smart Mobile Devices (SMDs) and different types of smart city application, performed to pursue several goals.

The second part deals with a partial offloading operation, taking into account the possibility of delegating towards a cloud infrastructure only a portion of the computation load. It is focused on the tradeoff between energy consumption and execution time, in a non-trivial multi-objective optimization approach. Furthermore, a utility function model developed from the economic field is introduced, in order to optimize the system. It takes into account a series of parameters related to the UMCC, showing that, when the network is overloaded, the partial offloading operation allows to achieve the target throughput values although the energy consumption and the computational time consumed in the partial offloading are lower than the resources consumed in the total offloading operation. In addition, the proposed UMCC framework and the partial computation offloading are applied to a vehicular environment for handling a real-time navigation application, so that the SMDs can exploit road side units and other neighbor devices forming clusters for delegating a shared application. It is shown that the clusterization allows to reduce the consumed energy in case of high traffic scenarios, optimizing the cluster size for different populations size and various offloading policies.

Finally, in the third part, the problem of Cell Association (CA) in a UMCC framework deals with the system as a community, thinking about improving the collective performance and not only the achievement of a single device. A probabilistic algorithm that uses biased-randomization techniques is proposed as an efficient alternative to exact methods, which require unacceptable levels of computational time to solve real life instances. This probabilistic algorithm is able to provide near-optimal solutions in real time, thus outperforming by far the solutions provided by existing greedy heuristics. Since this algorithm takes into account all users in the assignment process, it avoids the selfish or myopic behavior of the greedy heuristic and, at the same time, is able to quickly find near-optimal solutions for the allocation of the available resources.

Introduction

According to the flagship publication of the United Nations *World Urbanization Prospect*[1], more than one half of the world population is living nowadays in urban areas, and about 70% will be city dwellers by 2050. Furthermore, the world population is estimated to increase in the second half of the 21st century, while the urban areas are expected to absorb all the predicted growth and to draw in some of the rural population. The United Nations report predicts that, by mid-century, there will be 27 *megacities*, with at least 10 million population, while at least half of the urban growth in the coming decades will occur in small cities with less than 500,000 people, envisioning therefore that cities, big or small, are becoming a determining shift in the organization of human society. Cities and megacities are predicted to magnify problems such as difficulty in waste management, scarcity of resources, air pollution, human health concerns, traffic congestion, and inadequate, deteriorating and aging infrastructures.

Concurrently with such urbanization effect, an extraordinary phenomenon concerning the Information and Communication Technology (ICT) is happening: smart mobile devices are becoming an essential part of human life and the most effective and convenient communication tools, not bounded in time and place. According to the Cisco *Visual Networking Index*[2], the number of mobile-connected devices has already overtaken the number of people in the world, and by 2018 it will be over 10 billion, including Machine to Machine (M2M) modules. Overall mobile data traffic is expected to have nearly an 11-fold increase in the next five years.

Urbanization tendency and smart mobile expansion are going to reach a relevant convergence point through the concept of smart city, an icon of a sustainable and livable city, projecting the ubiquitous and pervasive computing paradigms to urban spaces, focusing on developing city network infrastructures, optimizing traffic and transportation flows, lowering energy consumption and offering innovative services. It is through ICT that smart cities are truly turning *smart* [1], in particular by means of the exploitation of smart mobile devices, forming together with cloud computing the Mobile Cloud Computing (MCC), since, as suggested by Michael Batty, *to understand cities we must view them not simply as places in space but as systems of networks and flows* [2].

In this context a new urban framework, named UMCC, is introduced in this thesis. It can

[1] http://esa.un.org/unpd/wup/

[2] http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

be thought as the technological nervous system allowing the networks and flows of the city for achieving a better urban way of life. By means of the UMCC framework, and considering various configurations of clouds and networks, data storage and processing can be dynamically delegated to resource-rich devices, thus shortening execution time, extending battery life and exploiting the possibility to preserve data in the cloud.

The UMCC framework can effectively support a smart city *vision*, gathering, collecting and processing data in real time, aiming to take advantage of the most advanced communication technologies, to hold up added-value services for the administration of the city and for the citizens. The resources optimization, within the UMCC framework, can be driven by purposely defined cost functions, including throughput, energy efficiency, latency and computing performance. The challenges and the exploiting opportunities of the UMCC are discussed in relation to smart city solutions, highlighting the features that can affect the Quality of Service (QoS) of various types of smart city related applications.

Original Contributions

In this dissertation, innovative techniques and methodologies aimed to enhance the performance of MCC applied to a smart city are proposed and investigated. In particular the UMCC, a global framework that can be adapted depending on the optimization objectives is introduced, highlighting the features that can affect the QoS of various types of smart city-related applications. Furthermore, various optimization techniques, based on opportunely defined cost or utility functions, are presented and inspected for the optimization of the resources in the UMCC. Specifically:

- a partial offloading tecnique is determined for optimizing time and energy consumption in a smart city HetNets scenario, where smart mobile devices are supposed to perform a distributed application;

- a utility function model derived from the economic world has been presented, aiming to measure the QoS, in order to choose the best access point in a HetNet for offloading part of an application on the MCC;

- a cluster-based optimization technique is proposed and utilized in a distributed computing resource allocation, exploiting resource sharing, in high density SMD environments.

During my Ph.D. course I had the opportunity to collaborate with the Internet Interdisciplinary Institute (IN3) at the Open University of Catalonia (UOC), where I contributed to develop new optimization heuristics for improving heterogeneous communication systems. The applied approach is based on the use of biased randomization techniques, which have been used in the past to solve similar combinatorial optimization problems in the fields of logistics, transportation, and production. This work extends the use of these biased randomization techniques to the field of smart cities and mobile telecommunications. Some numerical experiments contribute to the validation of the proposed approach.

The outcomes of this research stay are described in chapter 5.

Personal Publications

[P_1] D. Mazza, D. Tarchi, and G. Corazza, ”A Unified Urban Mobile Cloud Computing Offloading Mechanism for Smart Cities” *Communications Magazine, IEEE, Feb. 2016 (SUBMITTED)*

[P_2] D. Mazza, A. Pages, D. Tarchi, A. Juan, and G. E. Corazza, ”A biased-randomized algorithm for mobile cloud computing in smart cities,” *Systems Journal, IEEE, Oct. 2015 (SUBMITTED)*

[P_3] D. Mazza, D. Tarchi, and G. Corazza, ”A cluster based computation offloading technique for mobile cloud computing in a smart city scenario,” *in Proc. of IEEE Conference on Communication (ICC) 2016, Kuala Lumpur, Malaysia (ACCEPTED).*

[P_4] D. Mazza, D. Tarchi, and G. E. Corazza, ”A user-satisfaction based offloading technique for smart city applications,” *in Proc. of IEEE Globecom 2014, Austin, TX, USA, Dec. 2014.*

[P_5] D. Mazza, D. Tarchi, and G. E. Corazza, ”A partial offloading technique for wireless mobile cloud computing in smart cities, ” *in Proc. of 2014 European Conference on Networks and Communications (EuCNC), Bologna, Italy, Jun. 2014.*

Acknowledgments

This Ph.D. dissertation was carried out during three years of my life in which not only I have gained knowledge and accomplished an academic path, but also I had the opportunity to proof and enrich my personality. I would like to thank Prof. Giovanni Emanuele Corazza and Prof. Daniele Tarchi for their precious advice and support during my Ph.D. studies. They helped me during every step of my work with unmatched competence and patience.

A sincere thanks goes to Prof. Ángel Alejandro Juan Pérez and Prof. Adela Pagés Bernaus for giving me the chance to work with them at the DPCS IN3 Lab of the Open University of Catalonia: for their collaboration and for being always available to share projects and ideas and because my stay in Barcelona has been a unique experience for me, both professionally and personally.

I am also deeply grateful to Prof. Alessandro Vanelli Coralli, Coordinator of the Ph.D. Program in Electronics, Telecommunications, and Information Technologies, for making it possible to carry out this work.

A special acknowledgment goes also to all my colleagues that shared with me many important moments during my time at the University of Bologna - Roberta, Sergio, Vahid, Vincenzo - and at the Internet Interdisciplinary Institute IN3 in Barcelona - Aljoscha, Carlos, Jesica, Laura, Helena.

I also want to thank the reviewers of this thesis, Prof. Flavio Bonomi and Prof. Mohsen Guizani, for their many suggestions and help for greatly improving the quality of this work.

Infine, il piú grande ringraziamento va alla mia famiglia. Grazie a Gabriele, mio compagno di vita, per aver condiviso la mia esperienza e per avermi supportato e incoraggiato, ai miei genitori, per avermi infuso l'amore per lo studio, e a mio figlio Alessandro, che sempre mi ha dato la forza per continuare a dedicarmi alla ricerca.

Contents

Abstract **v**

Introduction **vii**

Original Contributions **ix**

Acknowledgments **xi**

I Urban Mobile Cloud Computing: a framework at the service of smart cities **1**

1 The UMCC architecture **3**

1.1 Introduction to the UMCC in a smart city scenario 3

1.2 Cloud Topologies 6

1.3 Types of RATs 7

1.4 Challenges of the UMCC 9

1.5 The UMCC model 11

1.6 Requirements of smart city applications 16

1.7 A utility based resource optimization approach 19

1.8 Conclusions 21

II Partial Offloading Optimization **23**

2 A Partial Computation Offloading Technique **25**

2.1 Introduction 25

2.2 System Model 28

2.3 Adaptive Offloading 30

2.4 Numerical Results 32

2.5 Conclusions 37

3 A User-Satisfaction Based Offloading Technique 39

3.1 Introduction 39

3.2 System Model 40

3.3 User-Satisfaction Based Utility Function 42

3.4 Numerical Results 44

3.5 Conclusions 47

4 A Cluster Based Computation Offloading Technique 49

4.1 Introduction 49

4.2 Application Scenario 50

4.3 The Partial Distributed Offloading Model 51

4.4 Cluster based optimization 54

4.5 Numerical Results 56

4.6 Conclusions 60

III Optimization considering a community of devices 61

5 A Social-Aware Biased-Randomized Algorithm for UMCC 63

5.1 Introduction 63

5.2 Problem Description 64

5.3 Optimal and Heuristic Solutions 70

5.3.1 Optimal Solution 70

5.3.2 The Greedy Heuristic 71

5.3.3 Biased-randomized Algorithm 72

5.4 Numerical Results 75

5.5 Analisys of Results 75

5.6 Conclusions 78

6 Conclusion and future works 87

6.1 Final Conclusions 87

6.2 Directions for Future Works 87

Bibliography 91

List of Tables

1.1 Summary of entities and relations in the UMCC - Involved features and requirements 14
1.2 Summary of smart city applications and Requirements 18

2.1 Offloading Parameters 29
2.2 Application Types 32

3.1 Values and fraction of computation and transmission 41
3.2 Reference values for QoS functions 43
3.3 Weight parameters of Utility Function 44

5.1 Throughput [Mbps] 83
5.2 Energy [W*s] 84
5.3 Time for application computation [s] 85

List of Figures

1.1 Mobile devices acting in the UMCC framework as (a) sensors, (b) nodes, (c) outputs. 5

1.2 Cloud topologies in the UMCC framework: centralized cloud, cloudlet and distributed mobile cloud. 8

1.3 The process of distributing and performing the application among different parts of the UMCC. 10

1.4 The utility function acts for distributing and performing the application in different parts of the Urban MCC. 21

2.1 The reference scenario with access nodes in HetNet for Mobile Cloud Computing . . 26

2.2 The offloading decision from the point of view of a single user. 28

2.3 Area in case of 500 and 5000 SMD connected, where the access points are positioned at point (0,0), (500,1000) and (0,1000), and the LTE station at (500,500) 33

2.4 Cost function behavior for Application 1 - Real Time Traffic Analysis 34

2.5 Cost function behavior for Application 2 - Mobile Video and Audio Communication 35

2.6 Cost function behavior for Application 3 - Mobile Social Networking 35

2.7 Energy for Application 3 - Mobile Social Networking 36

2.8 Time for Application 3 - Mobile Social Networking 36

3.1 Performance results in terms of average energy consumption with a variable number of SMDs. 46

3.2 Performance results in terms of average computation time with a variable number of SMDs. 47

3.3 Performance results in terms of average throughput with a variable number of SMD. 48

4.1 Realtime navigation with computation resource sharing, connection with the centralized cloud, the nearest cloudlet and near vehicles 56

4.2 Average latency vs the number of RSMDs, in a fluent traffic scenario 58

4.3 Average Energy vs the number of RSMDs, in a fluent traffic scenario 59

4.4 Average throughput vs the number of RSMDs, in fluent traffic scenario 60

5.1 The reference scenario with different node types in HetNet used by SMDs for offloading applications to UMCC 65
5.2 Scheme of the biased-randomization approach. 73
5.3 Comparison of the average throughput 76
5.4 Average performance in terms of throughput, energy consumption and latency for a variable number of devices 77
5.5 Cell Association in case of 500 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color. 78
5.6 Cell Association in case of 1000 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color. 79
5.7 Cell Association in case of 2000 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color. 80
5.8 Cell Association in case of 5000 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color. 81

6.1 Allocation of mobile devices with random positions to RATs 88
6.2 Random position of the devices in a target time 89

Glossary

3G short form of third generation, is the third generation of mobile telecommunications technology. This is based on a set of standards used for mobile devices and mobile telecommunications use services and networks that comply with the International Mobile Telecommunications-2000 (IMT-2000) specifications by the International Telecommunication Union. 3G finds application in wireless voice telephony, mobile Internet access, fixed wireless Internet access, video calls and mobile TV. 3G telecommunication networks support services that provide an information transfer rate of at least 200 kbit/s. Later 3G releases, often denoted 3.5G and 3.75G, also provide mobile broadband access of several Mbit/s to smartphones and mobile modems in laptop computers. This ensures it can be applied to wireless voice telephony, mobile Internet access, fixed wireless Internet access, video calls and mobile TV technologies. 9

femtocell a small, low-power cellular base station, typically designed for use in a home or small business. A broader term which is more widespread in the industry is small cell, with femtocell as a subset. It connects to the service providers network via broadband (such as DSL or cable); current designs typically support four to eight active mobile phones in a residential setting depending on version number, and eight to 16 active mobile phones in enterprise settings. A femtocell allows service providers to extend service coverage indoors or at the cell edge, especially where access would otherwise be limited or unavailable. 7, 9

HSPA+ Evolved High-Speed Packet Access, is a technical standard for wireless, broadband telecommunication. HSPA+ enhances the widely used WCDMA (UMTS) based 3G networks with higher speeds for the end user that are comparable to the newer LTE networks. HSPA+ was first defined in the technical standard 3GPP release 7 and expanded further in later releases. 7

IaaS Infrastructure as a Service. The IaaS provider offers you raw computing, storage, and network infrastructure so that you can load your own software, including operating systems and applications, on to this infrastructure. This scenario is equivalent to a hosting provider provisioning physical servers and storage and lets customers install their own OS, web services, and database applications over the provisioned machines. Amazon is arguably the first major proponent of IaaS through its Elastic Computing Cloud (EC2) service. It permits to rent servers with a certain CPU speed, memory, and disk capacity along with the OS and applications that a need to have installed on them. However, customers can also install their own OSs (or no OS) and applications over this server infrastructure. Scaling and elasticity are under the responsibility of the customer. CPU time, storage space, and network bandwidth (related to data movement) are some of the resources that can be billed on a usage basis. 1, 6

IEEE 802.11 a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6, 5, and 60 GHz frequency bands. They are created and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802). The base version of the standard was released in 1997, and has had subsequent amendments. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand. 7

LTE an abbreviation for Long-Term Evolution, commonly marketed as 4G LTE, is a standard for wireless communication of high-speed data for mobile phones and data terminals. 7, 9, 27, 32, 44, 45, 74

PaaS Platform as a Service. Unlike the fixed functions offered by SaaS, it provides a software platform on which users can build their own applications and host them on the PaaS provider's infrastructure. The software platform is used as a development framework to build, debug, and deploy applications. It often provides middleware-style services such as database and component services used by applications. PaaS is a true cloud model in which applications do not need to worry about the scalability of the underlying platform (hardware and software). When enterprises write their application to run over the PaaS provider's software platform, the elasticity and scalability is guaranteed transparently by the PaaS platform. 1, 6

picocell a small cellular base station typically covering a small area, such as in-building (offices, shopping malls, train stations, stock exchanges, etc.), or more recently in-aircraft. In cellular networks, picocells are typically used to extend coverage to indoor areas where outdoor signals do not reach well, or to add network capacity in areas with very dense phone usage, such as train stations or stadiums. Picocells provide coverage and capacity in areas difficult or expensive to reach using the more traditional macrocell approach. 7, 9

SaaS Software as a Service. It is a series of hosted service offered from a cloud provider through a network connection. It can be used instead of desktop and server products. The interface to the software is usually through a web browser. SaaS saves the complexity of software installation, maintenance, upgrades, and patches (for example, for security fixes), because the software is managed centrally at the SaaS provider's facilities. Also, the SaaS provider can provide this service to multiple customers and enterprises, resulting in a multitenant model. The pricing of such a SaaS service is typically on a per-user basis for a fixed bandwidth and storage. Monitoring application-delivery performance is the responsibility of the SaaS provider. Salesforce.com is an example of a SaaS provider. The company was founded to provide hosted software services, unlike some of the software vendors that have hosted versions of their conventional offerings. 1, 6

WiFi a local area wireless computer networking technology that allows electronic devices to connect to the network, mainly using the 2.4 gigahertz (12 cm) UHF and 5 gigahertz (6 cm) SHF ISM radio bands.The WiFi Alliance defines WiFi as any wireless local area network (WLAN) product based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards. However, the term WiFi is used in general English as a synonym for WLAN since most modern WLANs are based on these standards. 7, 9, 27, 32, 44, 45, 75

WiMAX Worldwide Interoperability for Microwave Access. It is a family of wireless communications standards initially designed to provide 30 to 40 megabit-per-second data rates, with the 2011 update providing up to 1 Gbit/s for fixed stations. The name WiMAX was created by the WiMAX Forum, which was formed in June 2001 to promote conformity and interoperability of the standard. The forum describes WiMAX as a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL. IEEE 802.16m or WirelessMAN-Advanced is a candidate for the 4G, in competition with the LTE Advanced standard. 7

Acronyms

AP Access Point. 12, 13, 23

CA Cell Association. vi, 40, 44, 61, 63, 66, 71, 88

CARS Cloud Assisted Remote Service. 7

CCI Cloud Computing Infrastructure. 50–53, 57, 60

ETSI European Telecommunications Standards Institute. 16

FLP Facility Location Problem. 89

GAMS General Algebraic Modeling System. 75

HetNet Heterogeneous Network. v, ix, 1, 7, 9, 13, 23, 25, 27, 29, 31, 37, 39, 40, 50–52, 54, 55, 64, 66–68

ICT Information and Communication Technology. vii, 16, 87

IERC European Research Cluster on the Internet of Things. 15

IoT Internet of Things. 4, 16, 49, 60, 87

ITS Intelligent Transportation System. 56

M2M Machine to Machine. vii, 4

MANET Mobile Ad-hoc Network. 51

MCC Mobile Cloud Computing. vii, ix, 1, 3–5, 11, 23, 25, 27, 39, 40, 49, 50, 87

QoS Quality of Service. viii, ix, 1, 3, 10, 11, 16, 19–21, 23, 39–44, 49, 52, 63, 68, 82, 88

QSAP Quadratic Semi-Assignment Problem. 70, 75

RAT Radio Access Technology. 7, 9, 10, 21, 29–31, 40, 43, 64–70, 73–75, 77, 88, 89

RSMD Requesting Smart Mobile Device. 9–13, 15, 52–55

SenaaS Sensing as a Service. 7

SMD Smart Mobile Device. v, vi, ix, 1, 5, 6, 9, 12, 13, 23, 25, 27, 29–31, 39–46, 49–57, 59, 60, 64, 66–77, 88

SNR Signal to Noise Ratio. 15, 31, 57, 68

SPOF Single-Point-of-Failure. 6

UMCC Urban Mobile Cloud Computing. v–ix, 1, 3, 4, 7, 9, 11, 16, 18–21, 23, 27, 29, 50, 60, 61, 63, 64, 66, 82, 88

VCN Vehicular Cloud Network. 4

PART I

URBAN MOBILE CLOUD COMPUTING: A FRAMEWORK AT THE SERVICE OF SMART CITIES

Innovative designs of smart cities, aiming at realizing a vision where municipalities can use information and communications technologies to meet sustainability goals, boost local economies, and improve urban services, have been and are being adopted in the political agenda of many governments as a primary program, in a large number of developed and developing countries. This development is in line with the evolutionary trends in the Information Society [3].

The ever-growing demand for services from citizens and institutions, intending to make the cities *smarter* and improve the quality of life of the communities, has given a great boost to the conception of diverse wireless communication systems and has extended the envision of cloud architectures for providing infrastructures (IaaS), platforms (PaaS), and software (SaaS) [4,5] , offering computation, storage and network and going towards the integration with novel opportunistic communications as fog networking [6–8].

In order to interact with city services, MCC and wireless HetNets contribute in different and sinergic way for handling this smart city scenario, allowing ubiquitous and pervasive computing in a framework we called UMCC.

In this first part of the dissertation, the proposed UMCC system model is described and investigated. It takes into account a series of features related to HetNet's nodes, cloud architectures, SMDs' characteristics, in association with several types of application and goals - mobility, healthcare, energy and waste management, and so on. It can be employed in the optimization of the QoS's requirements related to the needs of citizens.

Smart cities applications are gaining an increasing interest among administrations, citizens and technologists for their suitability in managing the everyday life. One of the major challenges is managing in an efficient way the presence of multiple applications in this UMCC framework, in a Wireless HetNet environment, alongside the presence of a MCC infrastructure.

The content of the following chapter was extracted from publications [P1], [P2], [P3] and [P4]

CHAPTER 1

The Urban Mobile Cloud Computing Architecture

1.1 Introduction to the UMCC in a smart city scenario

The increasing urbanization level of the world population has driven the development of technology toward the definition of a smart city geographic system, conceived as a wide area characterized by the presence of a multitude of smart devices, sensors and processing nodes aiming to distribute intelligence into the city; moreover, the pervasiveness of wireless technologies has led to the presence of heterogeneous networks operating simultaneously in the same city area. One of the main challenges in this context is to provide solutions able to optimize jointly the activities of data transfer, exploiting the heterogeneous networks, and data processing, by using different types of devices. In this chapter, the UMCC framework is introduced, considering a mobile cloud computing model that describes the flows of data and operations taking place in the smart city scenario.

The challenges and the opportunities of exploiting the UMCC are discussed in relation to smart city solutions, highlighting the features that can affect the QoS of various types of smart city-related services.

The UMCC sprang from the MCC, that is gaining an increasing interest in the recent years, due to the possibility of exploiting both cloud computing and mobile devices for enabling a distributed cloud infrastructure [9]. Considering the peculiarity of the MCC, we can observe that, on one hand, the cloud computing idea has been introduced as an enabling technology for allowing remote computation, storage and management of information, and, on the other hand, the mobility skill allows to gain by the most modern smart devices and broadband connections for creating a distributed and flexible virtual environment. At the same time, the recent advances in the wireless technologies are defining a novel pervasive scenario where several heterogeneous wireless networks interact among them, giving users the ability to select the best network choosing among those present in a certain area. As a consequence, the development of the UMCC is introduced, gaining from both computing and wireless communication technologies. It is a challenging opportunity for the creation of smart city infrastructures, providing solutions fulfilling the urgent need for richer application and services, requested from citizens that, as mobile users, are facing many demanding tasks in relation to mobile device resources as battery life, storage and bandwidth.

The triple role of Smart Mobile Devices

By analyzing the technology systems underlying a smart city framework, mobile devices can be considered in a three-fold way, as illustrated in Figure 1.1:

- **Sensors:** They can acquire different types of data regarding the users and the environment, transmitting a large amount of information to the cloud in real time, by means of wireless communication systems. This is the underlying concept leading to the Internet of Things (IoT) network, profitably exploited to improve urban life, for instance for extracting descriptive and predictive models in the urban context of cities [10]. As well as the expansion of Internet-connected automation into a plethora of new application areas, IoT is also expected to generate large amounts of data from diverse locations, with the consequent necessity for quick aggregation of the data, and an increase in the need to index, store, and process such data more effectively.

- **Nodes:** They can form distributed mobile clouds where the neighboring mobile devices are merged for resource sharing, becoming integral part of the network. Furthermore, they can form Vehicular Cloud Networks (VCNs) offering content routing, security, privacy, monitoring, virtualization services [11], easy to be used for providing smart city services and applications, in particular for traffic and mobility control. This is the crucial concept of fog networking, where a collaborative multitude of users carry out a substantial amount of storage, communication, and data management in a collaborative way.

- **Outputs:** They can make the citizens aware of results and able to decide consequently, or become actuators without need of human intervention. This is the concept underlying M2M communications where computers, embedded processors, smart sensors, actuators and mobile devices acquire information and act in an autonomous way [12].

To perform this triple role, mobile devices have to become part of an infrastructure that is constituted by different cloud topologies and, at the same time, have to exploit heterogeneous wireless link technologies, allowing to address the different requirements of a smart city scenario. This infrastructure starts from the concept of MCC, where the cloud works as a powerful complement to resource-constrained mobile devices.

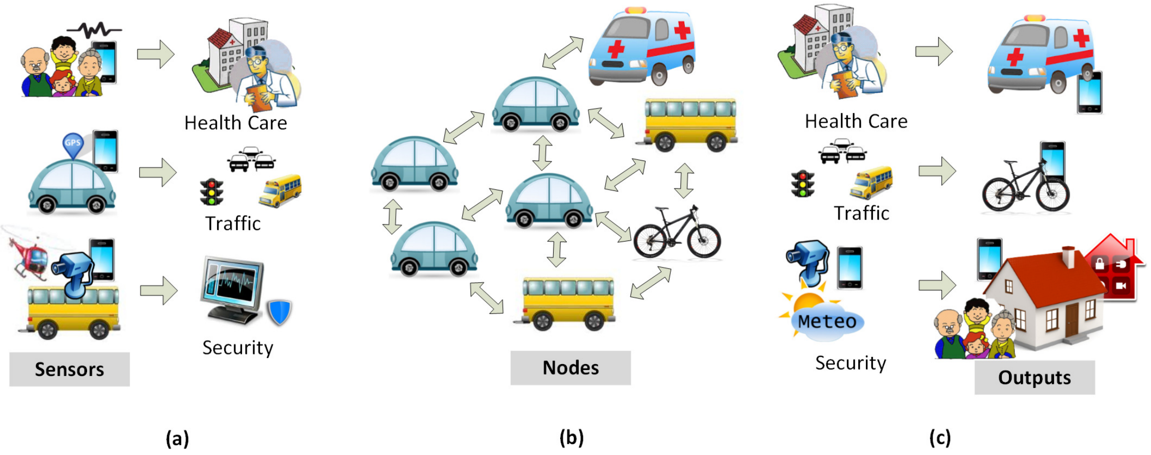


Figure 1.1: Mobile devices acting in the UMCC framework as (a) sensors, (b) nodes, (c) outputs.

The vision of MCC has increasingly become a source of interest, beginning from the early 2000s, when Amazon realized that a huge amount of space on their premises was underused. This awareness pushed toward the implementation of remote services, gaining by the presence of storage space and computing power and creating a cloud system. Alongside with the expansion of wireless technologies, the cloud computing has been integrated through a broadband system, exploiting the opportunity of working in mobility. The SMDs, then, can use MCC devolving demanding tasks and referring to it for data storage.

Computation Offloading

The strategy allowing to delegate to one or more cloud computing elements storage and computing functions is commonly called *cyberforaging* or *computation offloading*. It allows to tackle with the limited battery power and computation capacity of the SMDs, and plays a key role in a smart environment where wireless communication is of utmost relevance, particularly in mobility and traffic control domains [13]. If the storage is one of the most common and legacy activities that can be delegated to a remote cloud infrastructure, recently, thanks to modern programming paradigms, it is possible to allot even only a part of the computation load to a remote unit. This allows users to optimize the system performance by offloading only a fraction of the application to be computed, or distributing the application among different cloud structures. Offloading is an effective network congestion reduction strategy to solve the overload issue compared to scaling and optimisation [14]. It enables network operators to reduce the congestion in the cellular networks, while for the end-user it provides cost savings on data services and higher bandwidth availability.

1.2 Cloud Topologies

In relation to the SMD's roles previously described, we take into account various cloud topologies. This is a different categorization with respect to the common taxonomy used for cloud computing - SaaS, PaaS and IaaS. It looks on the different interaction among the nodes constituting the cloud, instead of the services provided by the cloud itself, so we can distinguish among centralized cloud, cloudlet, distributed mobile cloud and a combination of all, as shown in Figure 1.2.

Centralized Cloud

A centralized cloud provides the citizens to interact remotely, e.g., for accessing to open data delivered by the public administrations. It refers to the presence of a remote cloud computing infrastructure having a huge amount of storage space and computing power, virtually infinite, offering the major advantage of the elasticity of resource provisioning. The centralized cloud infrastructure is often used for delivering the computing processes to remote clusters, owing a higher computing power, and/or for storing big amount of data. The centralized cloud allows to reduce the computing time by exploiting powerful processing units, but it could suffer from the distribution latency, due to the data transfer from the users to the cloud and *vice versa*, the congestion, due to the multiple users exploitations, and the resiliency, due to the presence of a single performing infrastructure leading to the Single-Point-of-Failure (SPOF) issue.

Cloudlet

One of the main drawback of the centralized cloud is the great distance between the mobile devices, requesting services, and the clusters, performing computation in the cloud. Even if the SPOF issue is often resolved by implementing mirroring or redundancy solutions, the big distance that may occur between users and centralized clouds can be better addressed by means of the introduction of cloudlets, representing small clouds installed in proximity of the users. Furthermore, the inclusion of cloudlets allows a most appropriate sizing depending on the number of contemporary requests of the users.

Cloudlets are fixed small cloud infrastructures installed between the mobile devices and the centralized cloud, limiting their exploitation to the users in a specific area. Their introduction allows to decrease the latency of the access to cloud services by reducing the transfer distance at the cost of using smaller and less powerful cloud devices.

Distributed Mobile Cloud

A third configuration can address the issue of non persistent connectivity, whereas both the previous concepts must assume a durable state of connection. In a distributed mobile cloud the neighboring mobile devices are pooled together for resource sharing [15]. An application from a mobile device can be either processed in a distributed and collaborative fashion on all the mobile devices or handled by a particular mobile device that acts as a server.

The possibility of implementing a distributed mobile cloud infrastructure has become a reality since the introduction of smarter and powerful mobile devices, e.g., smartphones, tablet, phablet, having the ability, even if limited, of computing and storaging. Moreover, it has to be noted that their number is still increasing, leading to a pervasive presence and allowing to form a *cloud* of pervasive distributed devices that can interact among them. This *fog network* architecture uses one or a collaborative multitude of end-user clients or near-user edge devices to carry out a substantial amount of storage (rather than stored in centralized clouds), communication, and control, configuration, measurement and management [7,8]. It can be seen as the *fog layer* that encapsulates phisical objects - equipped with computing, storage, networking, sensing, and/or actuating resources - and constitutes a piece of a wider Cloud Assisted Remote Service (CARS) architecture [16], a geographically distributed platform that connects many billions of sensors and things, and provides multitier layers of abstraction of sensors and sensor networks, enabling the Sensing as a Service (SenaaS).

Combination of different topologies

The proposed framework foresees the joint exploitation of the three aforementioned topologies. As outlined before, they are characterized by different features, leading to a different usage depending on the scenario. Hence, a joint exploitation could steer to a more efficient usage aiming to achieve the performance goals of a certain application. As it will be better specified below, a smart city scenario is characterized by the presence of a lot of different applications, each one with different characteristics and requirements. An integrated UMCC framework composed by centralized clouds, cloudlets and distributed mobile clouds, as shown in Figure 1.2, allows to respect the application requirements with regard to other solution in a more efficient way.

1.3 Types of RATs

In order to connect the devices, different types of Radio Access Technologies (RATs) should be taken into consideration, providing a pervasive wireless coverage.

Multiple RATs, such as IEEE 802.11, mobile WiMAX, HSPA+, LTE and WiFi, must be integrated to form a HetNet. For enhancing the network capacity, generally there is an increasing interest in deploying relays, distributed antennas and small cellular base stations - picocells, femtocells, etc - indoors in residential homes and offices as well as outdoors in amusement parks and busy intersections. These new network deployments, comprised of a mix of low-power nodes underlying the conventional homogeneous macrocell network, by deploying additional small cells within the local-area range and bringing the network closer to users, can significantly boost the overall network

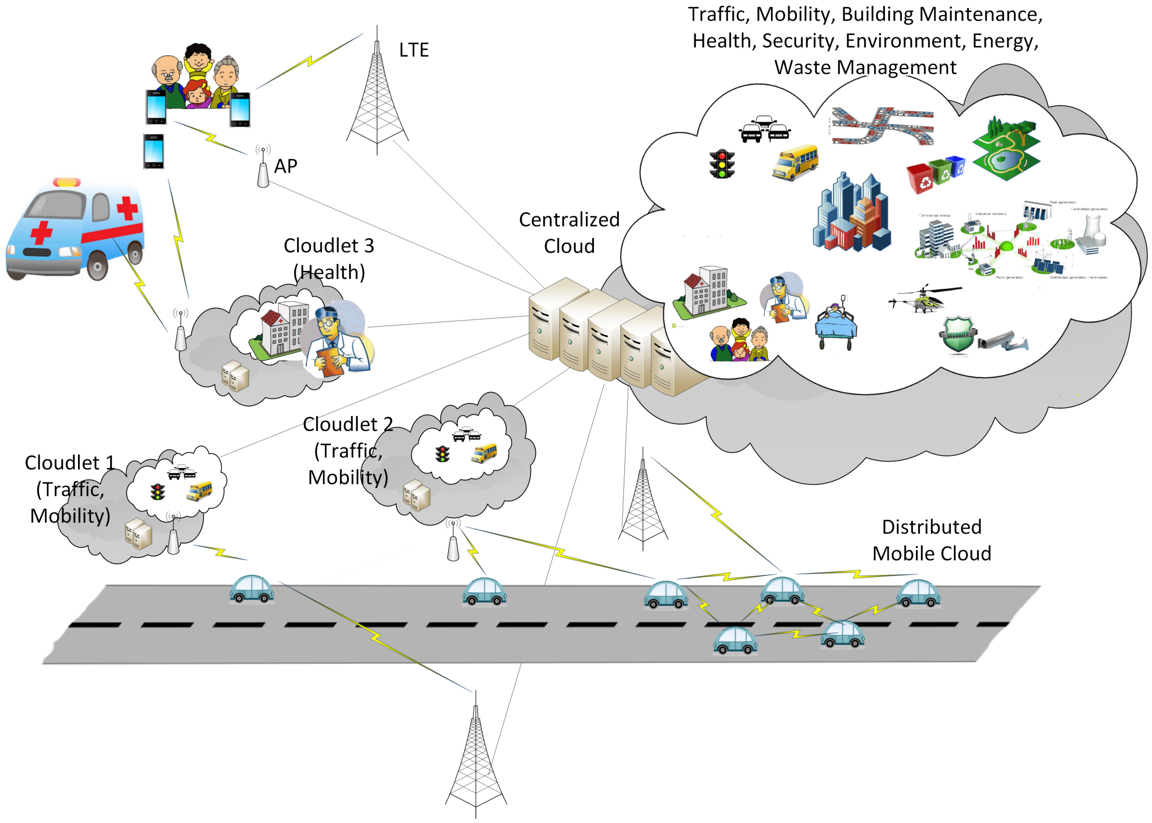


Figure 1.2: Cloud topologies in the UMCC framework: centralized cloud, cloudlet and distributed mobile cloud.

capacity through a better spatial resource reuse. Inspired by the attractive features and potential advantages of HetNets, their development have gained much momentum in the wireless industry and research communities during the past few years. The heterogeneous elements are distinguished by their transmit powers/coverage areas, physical size, backhaul, and propagation characteristics.

We can basically distinguish between two components, i.e., macrocells and small cells, where the former provide mobility while the latter boost coverage and capacity.

Macrocells

The distance between the access points (base stations of the macrocells) is usually higher than 500 m. Thanks to this type of base stations the environment is completely covered and the devices can move by minimizing the handover frequency. On the other hand, in macrocells the system suffers for channel fading and traffic congestion. This leads to a lack of stability, not allowing to reach very high data rate. The technology used for this type of cells refers to the cellular networks, e.g. 3G and LTE.

Small Cells

Small cells are characterized by low power radio access nodes, which have a cover range of about 100-200 m or less. We can distinguish between picocells, for providing hotspot coverage in public places - malls, airports and stadiums - without limits in terms of number of connected devices, and femtocells, for covering a home or small business area, available only for selected devices. Picocells and femtocells have been recently introduced as a way for increasing the coverage and maximize the resource allocation in LTE networks. We also consider WiFi access points as nodes with a small cover range (less than 100 m) which can typically communicate with a small number of client devices. However, the actual range of communication can vary significantly, depending on such variables as indoor or outdoor placement, the current weather, operating radio frequency, and the power output of devices.

1.4 Challenges of the UMCC

The UMCC approach foresees the definition of a scenario where smart city applications can exploit jointly the three topologies, as shown in Figure 1.3, by distributing and performing among the different parts composing the framework. The application requested by a particular SMD, signed as the Requesting Smart Mobile Device (RSMD), is partitioned and distributed among the different clouds using the available RATs. In the example of Figure 1.3, the application is divided among the centralized cloud, two cloudlets, and a distributed cloud formed by five devices. Furthermore, part of the application can be computed locally by the RSMD itself.

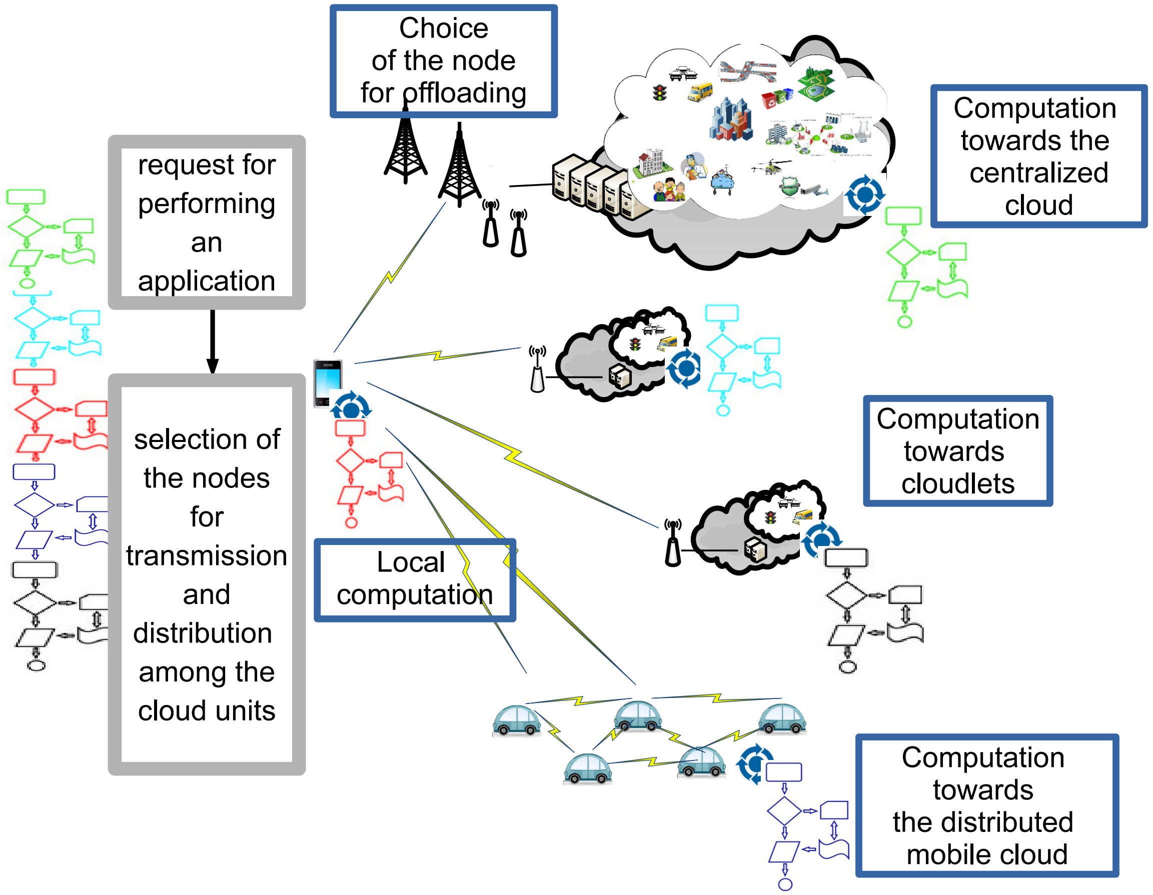

Choice of the node for offloading
request for performing an application
Computation towards the centralized cloud
selection of the nodes for transmission and distribution among the cloud units
Computation towards cloudlets
Local computation
Computation towards the distributed mobile cloud

Figure 1.3: The process of distributing and performing the application among different parts of the UMCC.

The main issue is that, for transferring data from the requesting mobile device to the selected cloud topology, a certain time is required. This mostly depends on some communication parameters of the selected RAT, such as the end-to-end throughput, the amount of users, the QoS management of a certain transmission technology between the user device and each type of cloud processing unit. Furthermore, in terms of energy consumption, it should be taken into account the tradeoff between the energy saved in offloading part of the application to the cloud and the energy spent in sending the data.

Hence, when a RSMD needs to select the clouds infrastructures to be used for computing the smart city application, we must focus on two main elements:

- the processing and storage devices - smart mobiles, *per se* or together forming distributed mobile clouds, and cloud servers, constituting the cloudlets and the centralized cloud;
- the wireless transmission equipments, - different RAT nodes entailing diverse transmission

speeds in relation to their own channel capacity and to the number of linked devices.

In Figure 1.3, the UMCC framework is sketched by representing the functional flows of the architecture. Whenever a smart city application should be performed, the citizen within the UMCC can select among different MCC infrastructures, i.e., centralized clouds, cloudlets, and distributed mobile clouds, aiming to respect the requirements of the specific application depending on their features. The distribution depends on the application requirements, and the UMCC features; its optimization will be discussed in the Section 1.7.

Computation, storage, and tranmission features

The features of the selected processing and storage devices, considered *per se* or in a group forming cloud/cloudlets, are:

- *Processing Speed*: The processing speed corresponds to the performance speed of a device or a group of devices for processing the applications;
- *Storage Capacity*: The storage capacity corresponds to the amount of storage space provided by a device or a group of devices.

In the same time, the features of the transmission equipments to be taken into account are:

- *Channel Capacity*: The nominal bandwidth of a certain communication technology that can be accessed by a certain device;
- *Priority/QoS management*: The ability of a certain communication technology to manage different QoS and/or priority levels;
- *Communication interfaces*: The number of communication interfaces of each device, that impacts on the possibility of selection among the available heterogeneous networks.

1.5 The UMCC model

In this paragraph we focus on the different entities playing a role in the UMCC framework, describing the functions and the interactions among them. First of all, we are focusing on an application App requested by a RSMD, defined through the number of operation to be executed, O, the amount of data to be exchanged, D, and the amount of data to be stored, S. An application can be seen as a smart city service, that can be executed either locally or remotely by exploiting the cloud infrastructures. Furthermore, each application has many requirements regarding the levels of QoS. We have taken into account the following:

- the maximum accepted latency T_{app}, intended as the interval between the request of performing an application and the acquisition of its results by the RSMD,

- the minimum level of energy consumption E_{app}, that the RSMD necessarily uses for performing the application itself,
- the throughput η_{app}, intended as the minimum bandwidth that the application needs for being performed.

Hence, for highlighting the *App* dependence from the above measures, we can write:

$$App = App(O, D, S, T_{app}, E_{app}, \eta_{app}) \tag{1.1}$$

A foundamental entity acting in the system is the RSMD requesting the *App*, we named *Dev*, characterized by certain features that are involved in the offloading operation: the power to compute applications locally, P_l, the power used for transferring data towards clouds, P_{tr}, the power for idling during the computation in the cloud, P_{id}, the computing speed to perform locally the computation, f_l, and the storage availability, H_l. Furthermore, also the time-varying position of the device plays an important role in the system interactions. Hence, we can write:

$$Dev = Dev(P_l, P_{tr}, P_{id}, f_l, H_l, pos_{dev}(x, y)) \tag{1.2}$$

Focusing on the different types of cloud entities in our scenario, we considered a unique centralized cloud C_{cc} and various cloudlets C_{cl} characterized by their own computing speed to perform the computation, i.e., f_{cc} for the centralized cloud and f_{cl} for the cloudlets. Additionally, the storage availability H_{cl} of each cloudlet has to be taken in consideration, while the storage availability of the centralized cloud can be considered infinite, therefore not constraining in the interaction. Hence, we can write for the centralized cloud C_{cc}:

$$C_{cc} = C_{cc}(f_{cc}) \tag{1.3}$$

and for each cloudlet, considering also the influence of the position $pos_{cl}(x, y)$ of the in-built Access Point (AP), the end-to-end throughput η_{cl} provided by the AP itself, the maximum number of devices that it is possible to connect at the same time n_{cl}, and the range of action r_{cl}:

$$C_{cl} = C_{cl}(f_{cl}, H_{cl}, pos_{cl}(x, y), \eta_{cl}, n_{cl}, r_{cl}) \tag{1.4}$$

We are considering the system from the point of view of a single RSMD requesting to run an application, while the set of the other SMDs constituting the distributed cloud are providing a service for supporting the RSMD. Thus, the distributed cloud is a set of generic entities *MDs*, each characterized by its specific connectivity, computation and storage for the exchange of data, i.e. the computing speed f_{MD}, the storage availability H_{MD}, the position $pos_{MD}(x, y)$, the throughput η_{MD}, the number of devices that can be connected to each devicen_{MD}, and each range of action r_{MD}.

Thus, we can write for the generic device *MD*:

$$MD = MD(f_{MD}, H_{MD}, pos_{MD}(x, y), \eta_{MD}, n_{MD}, r_{MD}) \tag{1.5}$$

While the connection to the cloudlets can be made only through the unique AP that can be considered built-in in each cloudlet itself, and the connection of the distributed cloud to the RSMD can be made directly, the nodes of the HetNet offer different choices to connect towards the centralized cloud. Thus, for each involved node *Nod* constituting the centralized cloud, specifying that $pos_{Nod}(x, y)$ is the position of the node, η_{Nod} is the end-to-end throughput in bit per second between the user and the exploited node, n_{Nod} is the number of devices available to connect, and r_{Nod} is the range of availability of the node, we can write:

$$Nod = Nod(pos_{Nod}(x, y), \eta_{\text{Nod}}, n_{\text{Nod}}, r_{\text{Nod}}) \tag{1.6}$$

The Table 1.1 summarizes the entities and the characteristics above described. They are in a certain relationship due to some physical and logical constraints derived from the following considerations.

First, for distributing the computation of the application among the different types of clouds, the system has to evaluate which HetNet nodes, cloudlets, and SMDs are available. The availability is realized if the RSMD is in the range of action of a particular HetNet node, cloudlet or SMD and if these entities are not busy, i.e., if the number of devices connected to an entity n_{conn} is less than n_{Nod}, n_{cl}, or n_{MD}, dependently from the type of entity.

Thus, there are M available HetNet nodes *Nod* for offloading towards the centralized cloud, N cloudlets C_{cl} and K devices *MD*, able to share the computation in the distributed cloud. After the detection of the available entities (which are a total of $1 + M + N + K$, including the local node RSMD, which we consider for simplicity the node of index 0), the next step is to distribute, by means of all these entities, different percentages α_i of operations O, β_i of data D, and γ_i of memory S, to all the available nodes, cloudlets and devices, under the constraints:

$$\sum_{i=0}^{M+N+K} \alpha_i = 1 \tag{1.7}$$

and

$$\sum_{i=1}^{M+N+K} \beta_i = 1 \tag{1.8}$$

Alongside the computing capacity, it is possible to define a constraint regarding the storage availability of the cloudlets and the SMDs by means of the following equations, considering infinite the

Table 1.1: Summary of entities and relations in the UMCC - Involved features and requirements

Entity	Connectivity	Storage	Throughput	Energy	Time latency
Reference Equation		1.9	1.10	1.11	1.12
$App = App(O, D, S, T_{app}, E_{app}, \eta_{app})$	-	S	η_{app}	O, D, E_{app}	O,D, T_{app}
$Dev = Dev(P_l, P_{tr}, P_{id}, f_l, H_l, pos_{dev}(x, y))$	$pos_{dev}(x, y)$	H_l	-	P_l, P_{tr}, P_{id}, f_l	f_l
$C_{cc} = C_{cc}(f_{cc})$	-	-	-	f_{cc}	f_{cc}
$C_{cl} = C_{cl}(f_{cl}, H_{cl}, pos_{cl}(x, y), \eta_{cl}, n_{cl}, r_{cl})$	$pos_{cl}(x, y)$, n_{cl}, r_{cl}	H_{cl}	η_{cl}	f_{cl}, η_{cl}	η_{cl}, f_{cl}
$MD = MD(f_{MD}, H_{MD}, pos_{MD}(x, y), \eta_{\mathrm{MD}}, n_{\mathrm{MD}}, r_{\mathrm{MD}})$	$pos_{MD}(x, y)$, n_{MD}, r_{MD}	H_{MD}	η_{MD}	f_{MD}, η_{MD}	η_{MD}, f_{MD}
$Nod = Nod(pos_{Nod}(x, y), \eta_{\mathrm{Nod}}, n_{\mathrm{Nod}}, r_{\mathrm{Nod}})$	$pos_{Nod}(x, y)$, n_{Nod}, r_{Nod}	-	η_{Nod}	η_{Nod}	η_{Nod}

storage availability of the centralized cloud:

$$\gamma_i S \leq H_i \qquad \forall i \in \{0, 1, ..., i, ..., N{+}K\} \tag{1.9}$$

that stands for an upper limit of remotely used storage of a certain i-th cloud infrastructure.

A constraint related to the application's requirements η_{app} involves the throughput of the entities designated for the offloading: the overall throughput offered by the selected devices should be higher than the minimum throughput requirement of a certain application:

$$\sum_{i=1}^{M+N+K} \eta_i \geq \eta_{app} \tag{1.10}$$

From the point of view of the RSMD, the energy spent for offloading the application can be written as the sum of the energy spent to perform locally a part of the task, plus the energy spent by the RSMD for the transmission of data to the clouds, plus the energy spent during the idle period when the computation is being offloaded. Hence, the restriction related to the requirement E_{app} leads to the following:

$$P_l \frac{\alpha_0 O}{f_l} + \sum_{i=1}^{M+N+K} P_{tr} \frac{\beta_i D}{\eta_i} + P_{id} \operatorname*{arg\,max}_{i=1,M+N+K} \left\{ \frac{\alpha_i O}{f_i} \right\} \leq E_{app} \tag{1.11}$$

In the same time, the total latency is the sum of the time for computing locally the α_0 percentage of computation, plus the times to transmit/receive data to/from the other computation units, plus the maximum of the time to compute in offloading. Hence, the restriction related to the requirement T_{app} leads to the following:

$$\alpha_0 \frac{O}{f_l} + \sum_{i=1}^{M+N+K} \frac{\beta_i D}{\eta_i} + \operatorname*{arg\,max}_{i=1,M+N+K} \left\{ \frac{\alpha_i O}{f_i} \right\} \leq T_{app} \tag{1.12}$$

Furthermore, the throughput η_i is related to the number of devices n_i connected to the i^{th} entity and the channel capacity BW_i as shown by the following representing the Shannon Formula:

$$\eta_i = \frac{BW_i}{n_i} \cdot \log_2 \left(1 + \frac{SNR_i}{d_i^2} \right) \tag{1.13}$$

where SNR_i is the Signal to Noise Ratio (SNR) of the related link and d_i the distance between the receiver and the transmitter. Thus, the optimization of the system consists in finding the values of α_i, β_i and γ_i that satisfy eqs. (1.7)-(1.13). This is a nontrivial optimization problem, but the complexity can be decreased with the introduction of some simplifications, as presented in the following chapters.

1.6 Requirements of smart city applications

There are many taxonomies trying to define smart city key areas, where social aims, care for environment, and economic issues are related and interconnected. The European Research Cluster on the Internet of Things (IERC) has identified in [17] a list of applications in different domains of IoT, including the smart city domain, showing the utmost strategic technology trends for the next five years. Moreover, the Net!Works European Technology Platform for Communications Networks and Services has issued a white paper [18] aiming to identify the major topics of smart cities that will influence the ICT environment. Furthermore, a relevant document aiming to categorize and define the different applications has been released by European Telecommunications Standards Institute (ETSI), where several application types have been specified focusing on their bandwidth requirements [19].

Taking into account all the relevant observations presented in these essays, we analyzed some particular smart city applications covering the areas of mobility, healthcare, disaster recovery, energy, waste management and tourism, in order to leverage the UMCC identifying the requirements which are related to the QoS.

Each application is defined through the service provided to the citizens, concerning the requirements in terms of throughput, energy consumption, time due to the transferring and computation processes, and number of users. In addition, for every application, the typical requirements of processing, data to exchange and storage have been established. The following list summarizes the definition of these requirements:

- *Latency*: The latency is defined as the amount of time required by a certain application between the event happens and the event is acquired by the system;
- *Energy Consumption*: The energy consumption corresponds to the energy consumed for executing a certain application locally or remotely;
- *Throughput*: The throughput corresponds to the amount of bandwidth required by a specific application to be reliably installed in the smart city environment;
- *Computing*: The computing corresponds to the amount of computing process requested by a certain application;
- *Exchanged data*: The exchanged data correspond to the amount of input, output and code information to be transferred by means of the wireless network;
- *Storage*: The storage corresponds to the amount of storage space required for storing the sensed data and/or the processing application;
- *Users*: The users correspond to the number of users for achieving a reliable service.

The QoS is a function of the previous requirements, where each one of these plays a role less or more important depending on the aims of the application. In the following list the considered application types are described by highlighting their technological requirements and characteristics, while in Table 1.2 the considered application types and the significance of their requirements are summarized.

Mobility All the components in an intelligent transportation system could be connected to improve transportation safety, relieve traffic congestion, reduce air pollution and enhance comfort of driving. The three-layered hierarchical cloud architecture for vehicular networks proposed by Yu et al. [20], where vehicles are mobile nodes that exploit cloud resources and services, can be considered as included in the UMCC framework. A real-time navigation with computation resource sharing, where the computation resources in the central cloud are utilized for data traffic mining, requests a minimal latency, since a ready response is needed. On the other hand, considering that the great part of mobile devices can be recharged directly from the car, energy consumption has to be considered only for pedestrians and bicycles. The necessary throughput, the computational load and the amount of data to exchange are high, whereas we can think the storage as a secondary requirement, unless for security recording.

Healthcare Intelligent and connected medical devices, monitoring physical activity and providing efficient therapy management by using patients' personal devices, could be connected to medical archives and provide information for medical diagnosis. We considered in the UMCC framework the typical architecture proposed by He et al. [21], requiring a full integration of the clinical devices and efficient processing of the collected data, considering, for instance, a cloudlet nearby the home of the patient. In this case there are relatively low requirements regarding energy consumption, throughput and number of users, whereas the requirements of latency, computation, exchanged data and storage are high, considering the complexity of the management algorithm, the video monitoring for remote diagnosis and the data to store for the personal record archive.

Disaster Recovery In [22] a disaster relief scenario is described, where people are facing with the destruction of the infrastructures and local citizens are asked to use their mobile phones to photograph the site. Also this case introduces the three-layered cloud described in the UMCC framework, requiring to transmit a lot of data using the cameras provided by the smartphones to reconstruct the disaster scene. In this case there are relatively low requirements regarding throughput, whereas it is important to have a quick response and to save the energy of the devices. There are a lot of computation to reconstruct the scene and a lot of data to exchange and to store. The number of users is variable.

Table 1.2: Summary of smart city applications and Requirements

	Requirements						
Application	latency	energy	throughput	computing	exchanged data	storage	users
Mobility	restrictive	variable	restrictive	high	high	variable	high
Healthcare	restrictive	non-restrictive	non-restrictive	high	high	high	low
Disaster Recovery	restrictive	restrictive	non-restrictive	high	high	high	variable
Energy	non-restrictive	non-restrictive	non-restrictive	high	high	high	high
Waste Management	non-restrictive	restrictive	non-restrictive	low	low	low	low
Tourism	non-restrictive	restrictive	non-restrictive	high	high	high	variable

Energy Energy saving can take advantage from the cloud basically thanks to smart grid systems, aimed to transform the behavior of individuals and communities towards a more efficient and greener use of electric power. Data fusion and mining, as well as scheduling and optimization, are critical in order to include the use of wireless communications to collect and exchange information about electric quality, consumption, and pricing in a secure and reliable fashion. By analyzing the specific aspects regarding the UMCC, if we consider an application where vehicles are involved in a smart grid system, we can suppose that a big data exchange is needed, there is a lot of computation and storage between a large number of users, whereas there are relatively low requirements regarding latency and throughput, and energy saving is not a problem for the devices involved.

Waste Management Automatically generated schedules and optimized routes which take into account an extensive set of parameters (future fill-level projections, truck availability, traffic information, road restrictions, etc.) could be planned not only looking at the current situation, but also considering the future outlook. A logistic solution that uses wireless sensors to measure and forecast the fill-level of waste and recycling containers could combine fill-level forecasts with an extensive set of collection parameters (e.g., traffic information, vehicle information, road restrictions) in order to calculate the most cost efficient collection plan. This smart plan would be automatically generated and accessed by the driver through a tablet[1]. We can expect non-restrictive requirements of latency and throughput, whereas resource-poor equipments have to be taken into consideration. The requirements related to data to be exchanged, load of computation, storage and number of users are not critical.

Tourism Augmented reality and social networks are the characteristics of applications that more take advantage from the cloud, that becomes also useful for mobile users sharing photos and video clips, tagging their friends in popular social networks like Twitter and Facebook. The cloud is effective also when mobile users require searching services, also using recognition techniques (using voice, images and keywords)[2]. We can expect not-restrictive requirements of latency and throughput, whereas resource-poor equipments have to be taken into consideration. There are a great amount of data to be exchanged, load of computation and storage and number of users are variable.

1.7 A utility based resource optimization approach

The requirements related to the applications, and the associated QoS, can be respected by optimizing the application partitioning and node/cloud association based on the features of the processing and storage devices and of the transmission equipments introduced in section 1.4. Moreover, it should be

[1] Example from http://www.enevo.com/
[2] Example from http://www.mtrip.com/augmented-reality/

noted that there is a correlation among the application requirements and the features of the UMCC equipments.

The latency suffered to perform a certain application can be seen as composed by the time needed for transferring the application data to the cloud computing infrastructure and the time needed for the cloud computation, therefore it is affected by the processing speed of the involved devices and by the channel capacity and the communication interfaces of the chosen transmission equipments. Furthermore, the energy spent to perform an application depends on the time needed for the data transfer, thus it depends on the channel capacity and the number of communication interfaces of the communication equipments; it has to be noticed that we have to consider also the energy spent by the user device while idles during the cloud computation, so that the processing speeds of the cloud devices are involved. The storage value of the application does not influence directly the performance of the system, but it can represent a limitation for the usability of a certain cloud infrastructure, as well as the number of potential users of the application.

A feasible optimization regarding the computing and the exchanged data requirements can be performed by operating on the complexity of the computability resources, considering also that, in partial offloading, the exchanged data could not be a predetermined value [23].

In this context, a utility function aiming to optimize the application-dependent QoS is proposed, acting as input for a procedure of partition and a node/cloud association, as shown in Figure 1.4. The utility function can be written as:

$$U = \sum_{i=1}^{N_{req}} \gamma_i f_i(\xi_i) \tag{1.14}$$

where N_{req} is the number of requirements for a certain smart city application, ξ_i stands for the i-th application requirement among those defined in section 1.6, and γ_i is a weight value, while $f_i(\cdot)$ corresponds to a specific utility function used for evaluate the respect of the i-th requirement.

In Figure 1.4, the functional blocks of the UMCC framework, based on a utility function optimization, are represented. On one hand, the smart city applications define specific requirements, while the cloud topologies in a certain scenario set their features. The utility function aims at selecting those cloud topologies that allow to respect the requirements by setting an optimized distribution of the application itself. The optimization of the partition and the node association will impact again on the UMCC features to be used by the other applications.

The maximization of the introduced utility function could be a nontrivial optimization problem, depending on the considered number of applications and devices acting in the selected scenario. In relation to the introduced requirements, the number of constraints could be reduced, and the complexity of the problem decreased by introducing opportunely defined sub-optimal solutions. All the utility and cost functions presented in the following chapters are particular cases of the Equation 1.14.

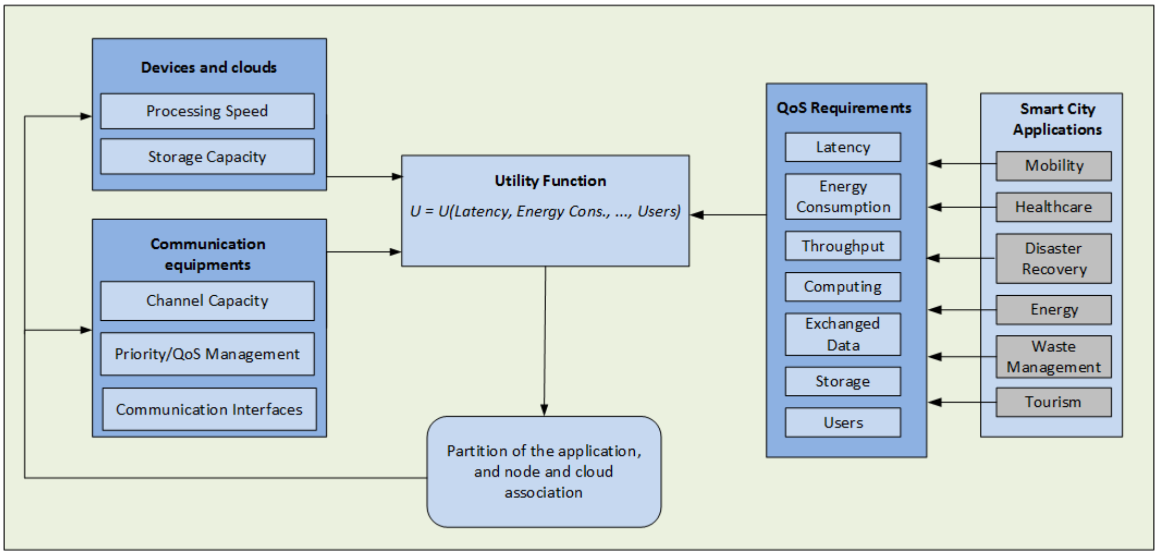


Figure 1.4: The utility function acts for distributing and performing the application in different parts of the Urban MCC.

In particular, in chapter 2 a cost function including the tradeoff between energy consumption of mobile devices versus the time to offloading data and to compute tasks on a centralized remote cloud server is provided, evaluating the optimal offloading fraction depending on the networks load; in chapter 3 and in chapter 5 we focused on a utility function representing the QoS degree perceived by the user, modeled as a sigmoid curve, that is a well-known function often used to describe QoS perception [24]; finally, in chapter 4 we focused on the computation offloading towards the distributed cloud, for offloading a real-time navigation application in a distributed fashion, aiming to minimize the execution time, since the devices are autonomous regarding the energy provision.

1.8 Conclusions

In this chapter we introduced the UMCC framework, a concept that supports the smart city vision for the optimization of the QoS of various types of smart city applications. The UMCC consists of different topologies of cloud and diverse types of RATs, that are used for offloading computation and share resources among the mobile devices. The QoS depends on the type of application, since it is affected by the defined requirements in a different way depending on the aims of the application itself. A cost function optimization approach is proposed, aiming to select the optimal partition level of the applications and the cloud infrastructures to be used for their computation. The optimization of the QoS is influenced by a big number of features, related to the choice of the distribution of data in the cloud units and the nodes used for the transmission, but the problem can be restricted considering some simplifications suggested by the aims and the domain of the application.

PART II

PARTIAL OFFLOADING OPTIMIZATION

A UMCC framework settled on an efficient wireless network allows users to benefit from multimedia services in an ubiquitous, seamless and interoperable way. In this context MCC and HetNets are viewed as infrastructures providing together a key solution for the major facing problems: the former allows to offload application to powerful remote servers, shortening execution time and extending battery life of mobile devices, while the latter allows the use of small cells in addition to macrocells, exploiting high-speed and stable connectivity in an ever grown mobile traffic trend. In order to fulfill the computation offloading efficiently, in the following chapters we explore techniques aiming to move the computing application towards the cloud, considering a non-trivial multi-objective optimization approach that takes into consideration the tradeoff between the energy consumption of the SMDs and the time for executing the application. The aim of the optimization is to find the percentage of application offloading that minimizes the proposed cost function, in such a way that only a part of the application is transfered and computed outside, whereas the rest of the application is computed locally.

The chapter 2 deals with the partial offloading technique in a *centralized cloud* scenario. The results show that exists a particular offloading percentage value fitting the system in case of simultaneously high data and network workload, differently from the simple *yes/no* offloading decision which would move the entire application or would perform it locally. In chapter 3 the partial offloading technique is applied using a utility function model arising from the economic world. It aims to measure the QoS, in order to choose the best AP in a HetNet for running the partial computation offloading. The goal is to save energy for the SMDs and to reduce computational time. In chapter 4 the proposed UMCC framework and the partial computation offloading are applied to a vehicular environment for handling a real-time navigation application. In this case, we consider also cloudlets and distributed clouds, so that the SMDs can exploit road side units and other neighbor devices for delegating a shared application.

The content of the following chapters was extracted from publications [P3], [P4], and [P5].

CHAPTER 2

A Partial Computation Offloading Technique

2.1 Introduction

Smart cities are considered a paradigm where wireless communication is an enhancing factor to make better urban services and improve the quality of life for citizens and visitors. In a smart city scenario several entities should be taken into consideration: the wireless infrastructure that allows data-exchange, the user devices, the sensing nodes, the machine devices, access points, one or more cloud infrastructures. Moreover, for delivering the requested services lots of data are exchanged among the citizens and the devices, and these data need also to be elaborated in order to give the correct information to the users. Thanks to various wireless communication technologies, users can move through different environments, indoor and outdoor, providing data to the cloud and receiving access services as browsing, video on demand, video streaming, information about location and maps. In this context, energy saving and performance improvement of the SMDs have been widely recognized as primary issues. In fact, the execution of every complex application is a big challenge due to the limited battery power and computation capacity of the mobile devices, especially in a smart environment where communication is considered a key to get better features in important areas such as mobility and transportation.

The exploitation of HetNet infrastructures together with the opportunity to delegate computation load to MCC, as shown in Figure 2.1, is an appealing connection achieving the aims of saving the SMD's power resource and executing the requested tasks in a faster way [23]. HetNets involve multiple types of low power radio access nodes in addition to the traditional macrocell nodes in a wireless network, reaching the major goal to enhance connectivity. On the other hand, MCC aims to increase the computing capabilities of mobile devices, to conserve local resources - especially battery

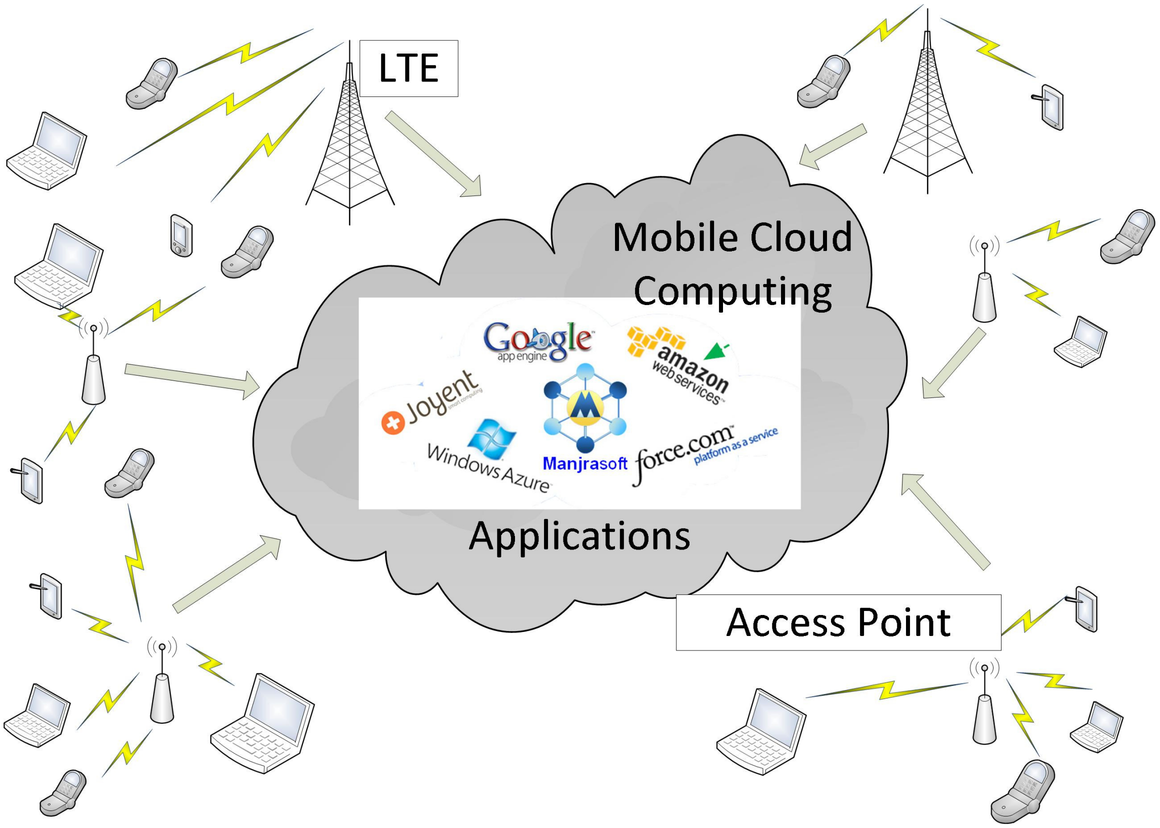


Figure 2.1: The reference scenario with access nodes in HetNet for Mobile Cloud Computing

charge - to extend storage capacity and to enhance data safety for making the computing experience of mobile users better [25].

The distributed execution (i.e., computation/code offloading) between the cloud and mobile devices has been widely investigated [9], highlighting the challenges towards a more efficient cloud-based offloading framework and also suggesting some opportunities that may be exploited. Indeed, the joint optimization of HetNets and distributed processing is a promising research trend [5].

Several works have already analyzed characteristics and capacity of MCC offloading, for example aiming to extract offloading friendly parts of codes from existing applications [26, 27]. Also, in [28] the key issues are identified when developing new applications which can effectively leverage cloud resources. Furthermore, in [29] a real-life scenarios, where each device is associated to a software clone on the cloud, has been considered, and in [30] a system that effectively accounts for the power usage of all of the primary hardware subsystems on the phone has been implemented, distinguishing between CPU, display, graphics, GPS, audio, and microphone.

In [31] an offloading framework, named Ternary Decision Maker (TDM), has been developed, aiming to shorten response time and reduce energy consumption simultaneously with targets of execution including on-board CPU and GPU in addition to the cloud, from the point of view of the single device. In addition, there are many studies that focus on whether to offload computation to a server, providing solutions related to a yes/no decision for the entire task at one time [32, 33], or studies that focus on optimization of the energy consumption in SMDs necessary to run a given application under execution time constraint [34].

Differently from the literature, we propose a partial offloading technique able to exploit the HetNets scenario and the presence of MCC devices - the UMCC framework - by optimizing the amount of partial offloading of the computational tasks depending on the number of devices connected to a network and their location with respect to the WiFi access points or LTE eNodeBs. The SMDs can exploit the partial data offloading to distribute high computational tasks among centralized servers and local computing.

In Figure 2.2 the workflow and the entities involved in the performance of a task are shown. From the point of view of a single user, the decision about offloading or not is taken on the basis of the following considerations:

- If the task is delegated to the cloud, the energy consumed by the mobile device is due to the data transfer - uploading data and downloading results - plus the energy consumed in an idle state during the outside computation - waiting for the results; the global time for having the task accomplished is related not only to the computational time but also to the transfer time for moving data from the mobile device to the cloud and *vice-versa*.

- If the task is computed locally, the energy consumed by the mobile device is due to the computation itself; the global time for having the task accomplished is related only to the computational ability of the SMD.

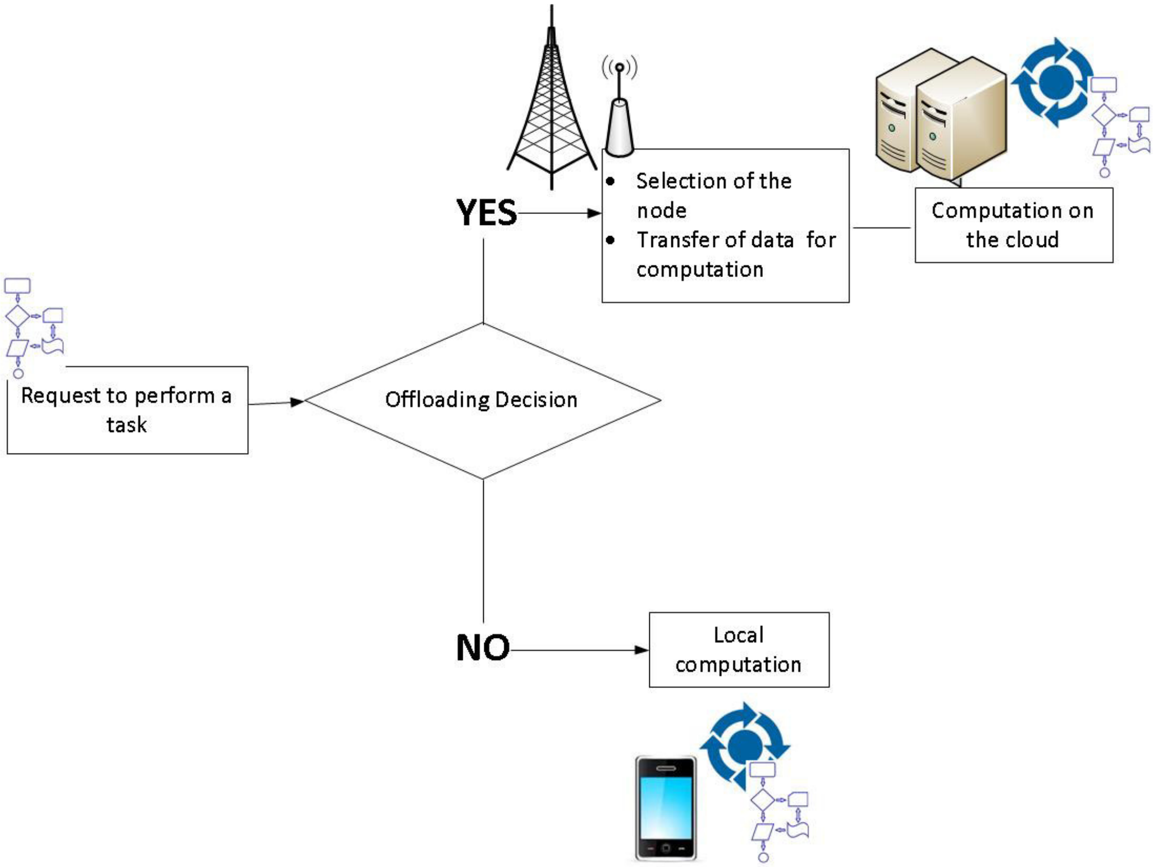


Figure 2.2: The offloading decision from the point of view of a single user.

In this chapter the optimization of the entire system is considered, not for a single device but for the whole community of devices, by taking into account partial offloading in a non trivial multi-objective optimization approach, where both energy consumption and execution time constraints are considered. A cost function including the tradeoff between energy consumption of mobile devices versus the time to offloading data and to compute tasks on a remote cloud server is provided, evaluating the optimal offloading fraction depending on the network's load. It can be exploited when the network is overloaded and the tasks request large amounts both of computation to perform and data to exchange.

2.2 System Model

The reference scenario is characterized by a urban area with a pervasive wireless coverage, where several mobile devices are interacting with a traditional centralized cloud and request for services from a remote data center, as illustrated in Figure 2.1. Alongside the presence of a pervasive wireless

Table 2.1: Offloading Parameters

symbol	meaning	unit of measure
P_l	power for local computing	W
P_{id}	power while being idle	W
P_{tr}	power for sending and receiving data	W
S_{md}	SMD's calculation speed	no. of instructions / s
S_{tr}	SMD's transmission speed	bit / s
S_{cs}	cloud server's calculation speed	no. of instructions / s
C	instructions required by the task	no. of instructions
D	exchanged data	bit

network, the system deals with many sensing and user terminals that generate and exploit a large amount of data. In order to connect the SMDs to the cloud and the data centers for delegating data for the computation, we take into account a simple categorization of the UMCC's trasmission entities, considering only two types of RATs forming the basic elements of the HetNets: macrocells and small cells. If, on one side, the strategy of delegating computation to the centralized cloud allows to exploit high performance computing centers, on the other side, it copes with the energy spent by the SMDs for transferring data. Similarly, the SMDs which compute locally the applications in a distributed approach, face with the energy issues due to the computation itself. In other words, the SMDs consumes energy both to delegate the application to the data centers and to compute it locally. Furthermore, both the speed to transmit data and the speed for local computation are related to the energy consumed by the SMDs. Thus, the offloading decision between offloading the application or computing it locally leads to a tradeoff. For this reason our model provides a cost function by resorting to a previously introduced model in [32, 33] which compares the energy used for a 100% offloading with the ones used to perform the task locally. The parameters used in the following are listed in Table 2.1.

In our scenario we suppose that the computation of a certain task requires C instructions. S_{md} and S_{cs} are, respectively, the speeds in instructions per second of the mobile device and the cloud server. Hence, a certain task can be completed in an amount of time equal to C/S_{md} on the device and C/S_{cs} on the server. On the other hand, let us suppose that D corresponds to the amount of bits of data that the device and the server must exchange for the remote computation, and S_{tr} is the transmission speed, in bit per second between the SMD and the access point; hence, the transmission of data lasts an amount of time equal to D/S_{tr}. In this case we consider that the transmission time is mostly due to the access network transfer, because the transfer rate of the backbone network can be considered as negligible due to the higher data rate. Moreover, we consider as negligible the transfer time from the access point to the user terminal because the amount of data in response to the elaboration in centralized server is little with respect to the data sent to the centralized server [32, 33].

Hence, it is possible to derive the energy for local computing:

$$E_l = P_l \times \frac{C}{S_{md}} \tag{2.1}$$

as the product of the power consumption of the mobile device for computing locally, P_l, and the time C/S_{md} needed for the computation. Similarly, it is possible to derive the energy needed for performing the task computation on the cloud as the energy used while being in idle for the remote computation plus the energy used to transmit the whole data from the SMD to the cloud:

$$E_{od} = P_{id} \times \frac{C}{S_{cs}} + P_{tr} \times \frac{D}{S_{tr}}, \tag{2.2}$$

where P_{id} and P_{tr} are the power consumptions of the mobile device, in watts, during idle and data transmission periods, respectively.

Similarly, it is possible to derive the time needed for the local computing as:

$$T_l = \frac{C}{S_{md}}, \tag{2.3}$$

and the time for the whole offloading computing as

$$T_{od} = \frac{C}{S_{cs}} + \frac{D}{S_{tr}} \tag{2.4}$$

In many applications, this approach is not efficient or feasible, and it is necessary to partition the application at a finer granularity into local and remote parts, which is a key step for offloading.

2.3 Adaptive Offloading

In this section, firstly, two equations are provided, to represent the energy used by a SMD to execute an application in partial offloading and to express the time needed to execute such application. Secondly, the impact of the traffic workload in the wireless network is taken into account, since the RATs and the number of SMDs entails the transmission speed of the the offloading data. Thirdly, a cost function is introduced, to evaluate the percentage of offloading which minimizes both energy and time.

In order to analyze the energy spent for offloading only a part of the application, we introduce the weight coefficients γ and δ, satisfying $0 \leq \gamma, \delta \leq 1$, representing respectively the percentage of the computational task and the percentage of the exchanged data for offloading. Then, the used energy of a single device E_{part_od} has been introduced, as the sum of the one spent to perform a part of the task locally plus the one spent to idle and transmit the other part of the task to the cloud:

$$E_{part\_od} = P_l \times \frac{(1-\gamma) \cdot C}{S_{md}} + P_{id} \times \frac{\gamma \cdot C}{S_{cs}} + P_{tr} \times \frac{\delta \cdot D}{S_{tr}} \tag{2.5}$$

Taking into account the same coefficients γ and δ used in Equation 2.5, we can calculate the time for the partial offloading T_{part_od} as the maximum between the time needed for computing the local part of the task and the time needed for the offloading, considering the two phases performed in the same time:

$$T_{part\_od} = \max\left(\frac{(1-\gamma) \cdot C}{S_{md}}; \quad \frac{\gamma \cdot C}{S_{cs}} + \frac{\delta \cdot D}{S_{tr}}\right) \tag{2.6}$$

The structure and the workload of the network are implicitly considered in Equation 2.5 and Equation 2.6. Now we are going to describe the effect on E_{part_od} and T_{part_od} due to the different RATs performing in the HetNets and to the amount of devices connected to this different RATs. The HetNet mainly consists of two components, macrocells and small cells, with different bandwidths BW. Since, for a single SMD, the speed rate of the data exchange S_{tr} is affected by the bandwidth of the node to which the SMD is connected, by the distance d from this node, and by the number n of the overall SMDs connected to the same node, S_{tr} can be written in an explicit way as:

$$S_{tr} = \frac{BW}{n} \cdot \log_2\left(1 + \frac{SNR}{d^2}\right) \tag{2.7}$$

where SNR is the SNR, typical parameter of the device.

In order to allow the evaluation of the offloading percentage, aiming to save energy and improve performance, the introduction of a cost function that can consider the minimization of both Equation 2.5 and Equation 2.6 for the entire set of SMDs is required. This is a non-trivial multi-objective optimization problem that we addressed by setting the cost function as a weighted sum of both the average values, with α and β coefficients with the constraints $0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$, N number of network's devices and E_l, T_l reference values representing average energy and time spent when the task is computed locally by a SMD:

$$F = \alpha \frac{\frac{1}{N}\sum_{k=1}^{N} E_{part\_od,k}(\gamma, \delta)}{E_l} + \beta \frac{\frac{1}{N}\sum_{k=1}^{N} T_{part\_od,k}(\gamma, \delta)}{T_l} \tag{2.8}$$

This cost function is based on a network centric approach in which a central entity is responsible for choosing values of the offloading percentage γ and δ after collecting informations about the SMDs' features. Furthermore, in the partial offloading procedure, γ and δ are bounds, because before a

Table 2.2: Application Types

Application	Computation	Data transmission	C	D	δ/γ
1 - Real time traffic analysis	High	Low	10^7	10^5	0.25
2 - Mobile Video and Audio Communication	Low	High	10^5	10^7	0.75
3 - Mobile Social Networking	High	High	10^7	10^7	0.50

task is executed it may require certain amount of data from other tasks [23]. Moreover, the weighted coefficients α and β are chosen at a main level to give a major importance to energy or time saving.

2.4 Numerical Results

During a partial offloading the amount of energy and time in Equation 2.5 and Equation 2.6 is affected by the percentage of computation and communication exchanged, represented respectively by the coefficients γ and δ. These are correlated each other, since the execution of a remote computation task requires a certain amount of input/output data to be exchanged. So we can consider the ratio $\frac{\delta}{\gamma}$ as a typical value, peculiar of a type of application. To summarize typical scenarios we have taken into account three kinds of applications represented in Table 2.2, according to the aims to analyze cases of a smart transportation system [35].

We considered a deployment area of $1000 \times 1000\ m^2$, where one LTE eNodeB with channel capacity equal to 100 MHz and three WiFi access point with channel capacities equal to 22 MHz are positioned to cover the entire area. Specifically, the access points are positioned at point (0,0), (500,1000) and (0,1000), and the LTE station at (500,500), as shown in Figure 2.3.

The values of S_{md}, P_{id}, P_{tr} and P_l are specific parameters of the mobile device. For example we utilized the values of an HP iPAQ PDA with a 400 MHz Intel XScale processor ($S_{md} = 400$) and the following values: $P_l \approx 0.9W$, $P_{id} \approx 0.3W$ and $P_{tr} \approx 1.3W$ [32].

The cost function coefficients, α and β, are equal to 0.5, aiming to give the same importance to both time and energy consumptions. In Figures 2.4, 2.5, and 2.6 the performance results of the cost function are represented for the three applications described in Table 2.2.

Figure 2.4 shows that, when a task requires high computation and low communication, i.e. Application 1, it is better to offload the task totally, no matter how many devices are connected to the network. In fact, the curves are overlapped and the cost function assumes the sames values for the same percentage γ.

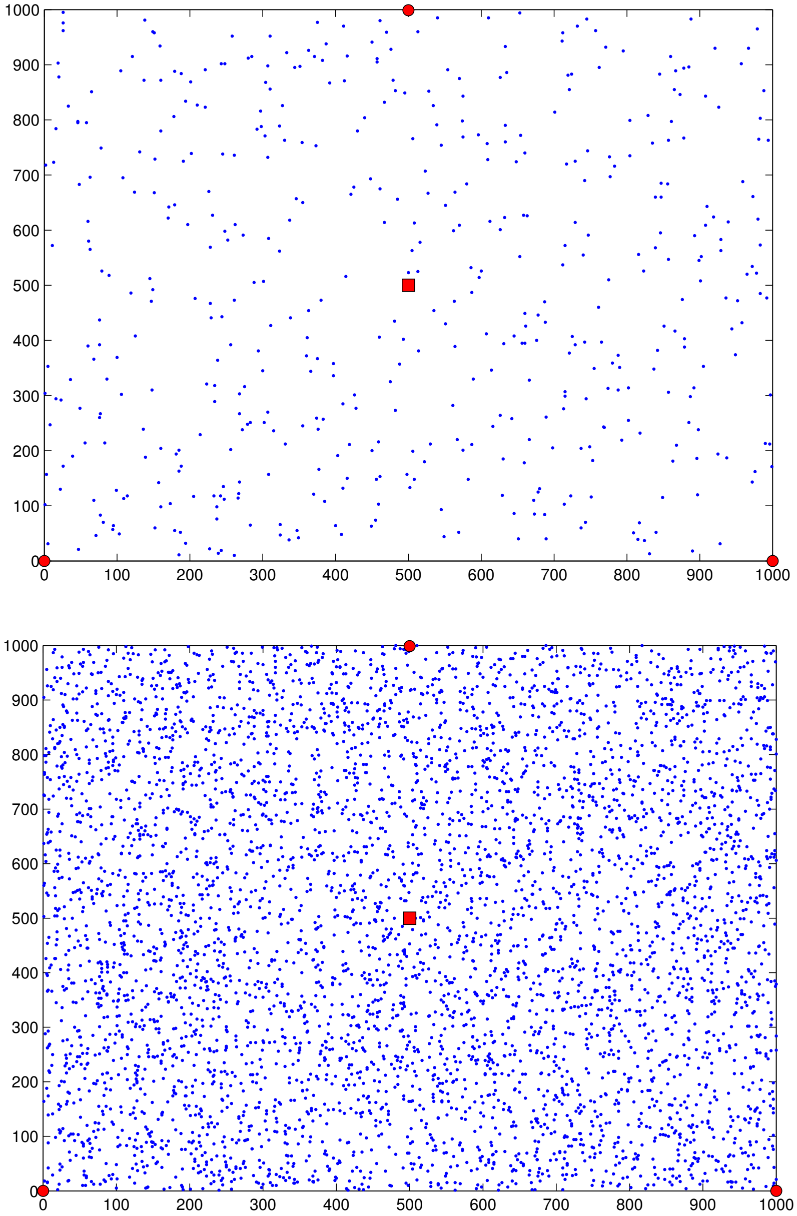


Figure 2.3: Area in case of 500 and 5000 SMD connected, where the access points are positioned at point (0,0), (500,1000) and (0,1000), and the LTE station at (500,500)

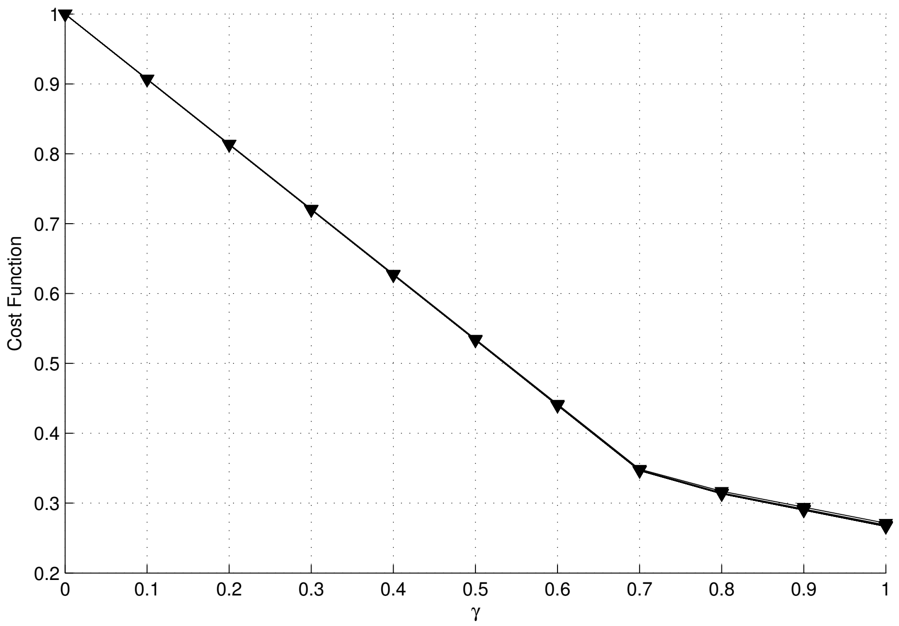


Figure 2.4: Cost function behavior for Application 1 - Real Time Traffic Analysis

On the other hands, Figure 2.5 shows that, when a task requires low computation and high communication, i.e. Application 2, it is better to compute the task locally. In this case a big number of connected devices affects the cost function in a negative way; it is possible to see that there is a minimum for $\gamma = 0$.

The most interesting case is shown in Figure 2.6. In fact, when the network is overloaded, tasks with both a large amount of computation to execute and data to exchange - as in Application 3 - are better performed for a specific value of γ. For example, in this case, the best performance is for $\gamma = 0.4$ when a population of 5000 devices is whithin the area, and $\gamma = 0.7$ for a population of 2000 devices. When the network is not overloaded, instead, as in cases of less devices within the area, it's better to perform the total offloading.

Finally, as shown in Figure 2.7 and Figure 2.8, we compare energy and time spent in the adaptive case with those spent for the local execution and the total offloading case to perform Application 3; for the adaptive algorithm we have considered the use of the optimized γ parameter following the previous analysis. While for the energy there is a compromise between the two boundary cases, the adaptive function allows the best performance considering time as the primary issue.

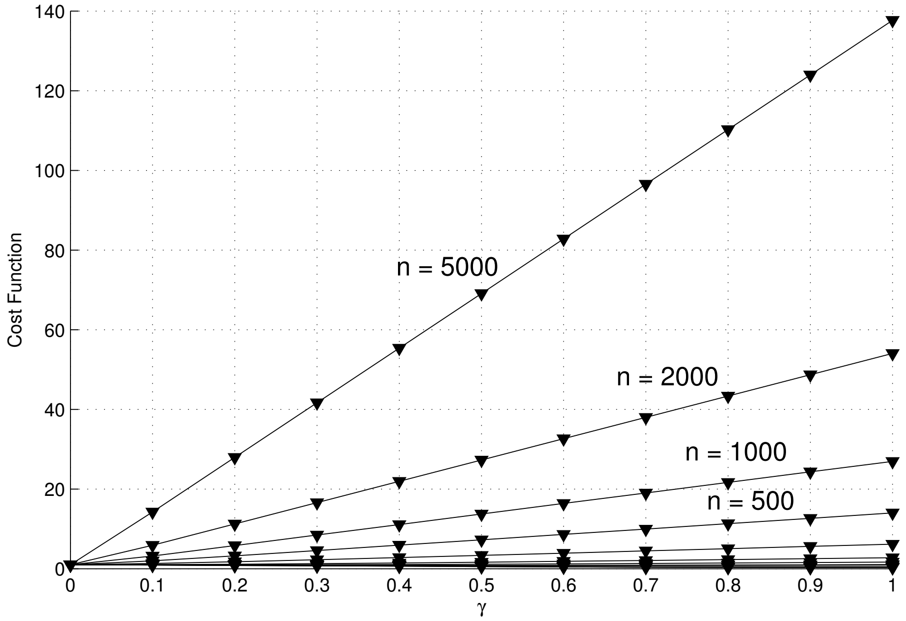


Figure 2.5: Cost function behavior for Application 2 - Mobile Video and Audio Communication

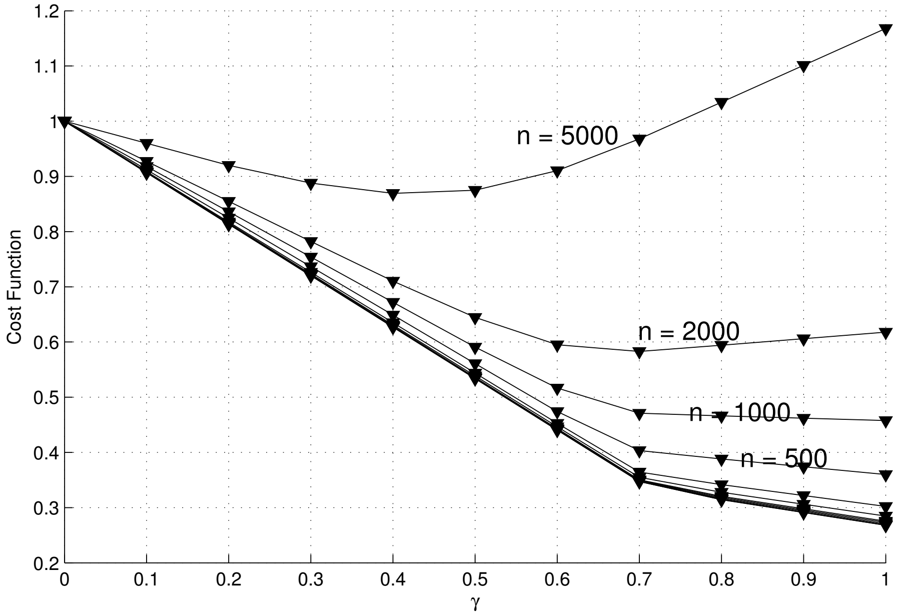


Figure 2.6: Cost function behavior for Application 3 - Mobile Social Networking

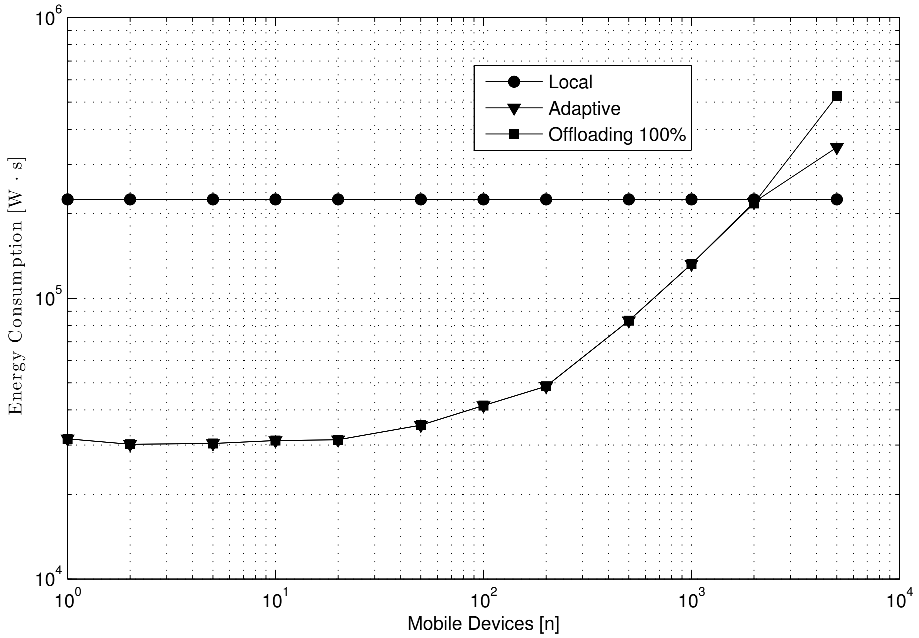


Figure 2.7: Energy for Application 3 - Mobile Social Networking

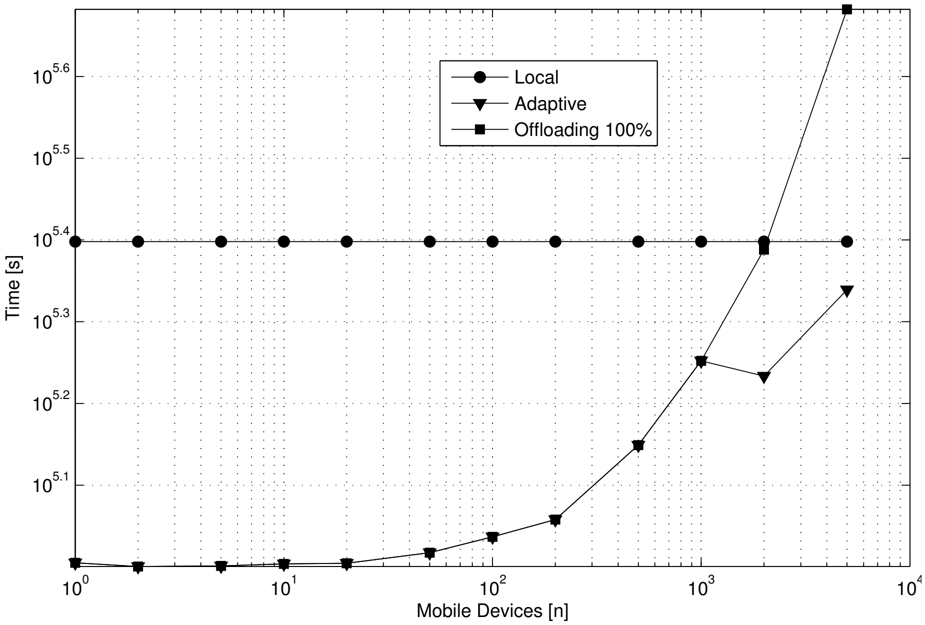


Figure 2.8: Time for Application 3 - Mobile Social Networking

2.5 Conclusions

In this chapter, a cost function has been defined for optimizing contemporaneously time and energy consumption in a scenario where smart mobile devices are supposed to perform an application; the aim was to optimize the amount of computation performed locally and remotely. The remote execution is faster and can relieve mobile devices from the correlated energy consumption, but it involves data exchange with the cloud server, spending time and energy to transmit, depending also from the load of the HetNet. The cost function is proposed to evaluate the percentage of application to offload for time and energy optimization. The results show that for applications requesting both high execution work and data exchange a particular value of this percentage, depending on the number of devices, optimize the performance.

CHAPTER 3

A User-Satisfaction Based Offloading Technique for Smart City Applications

3.1 Introduction

New user's needs cause a major boost of wireless communication techniques employed in smart cities, as well as a rapid growth and diffusion of enhancing technologies. To achieve the goal of interacting with city services, allowing to simplify everyday life, MCC and HetNets become together the killer applications for resolving the significant facing problems: the former for offloading application to powerful remote servers, shortening execution time and extending battery life of mobile devices, the latter for exploiting high-speed and stable connectivity in an ever grown mobile traffic trend, allowing the use of small cells in addition to macrocells [23]. In such a scenario users can access to remote resources without interruption in time and space.

In this context, energy saving and performance improvement of SMDs have been widely recognized as primary issues. In fact, the execution of every complex application is a big challenge due to the limited battery power and computation capacity of the mobile devices [25]. The distributed execution between the cloud and mobile devices has been widely investigated [9], highlighting the challenges towards a more efficient cloud-based offloading framework and also suggesting some opportunities that may be exploited. Indeed, the joint optimization of HetNets and distributed processing is a promising research trend [5]. We envision that the success of HetNets, jointly with MCC, would ultimately depends on user satisfaction, which in turn relies on saving energy and computing application quickly. Identifying the relevant QoS for each of the diverse application types and distinguishing the variation of user satisfaction related to the QoS is a research challenge [36].

Various mobile data offloading policies are proposed in the literature where the partial offloading of data to the network infrastructure is performed according to the variations of the network conditions and the operator strategies [37]; however, a model related to user satisfaction regarding battery saving and speed of computation is not already taken in consideration.

In this chapter we propose a utility function model that takes into account a series of parameters related to HetNet's nodes, SMDs' characteristics and types of performed application. We categorize applications in different classes, and consider for each application type the amount of data and computation transferred to the MCC and how the HetNet traffic load affects the power consumption of the SMDs and their execution time. The opportunity to move to the cloud a portion of the computing application is taken into account, because the decision whether moving the computation tasks of mobile applications from the local SMDs to the remote cloud involves a tradeoff between energy consumption and computational time [38].

The proposed utility function takes into account the QoS parameters in terms of throughput, amount of energy used by the SMDs and time spent to execute the application. Furthermore, it acts as input for a CA procedure aiming to select the best access point for respecting the system requirements. The user-satisfaction CA algorithm is compared with a legacy algorithm that foresees the connection with the nearest access point. The results show that when the network is overloaded, the algorithm based on the proposed utility function offers a better service with respect to the nearest-node technique, since the average throughput is stable, allowing less outage of connectivity and reduced values of average energy and average time in comparison to the nearest-node algorithm.

3.2 System Model

The system model we are focusing on relies on the results of the chapter 2. We consider a reference scenario analogous to Figure 2.1, but, in this case, various SMDs are requesting three different types of application, with a casual uniform distribution. Each type is partially offloaded giving the optimization problem investigated previously. The SMDs are interacting with a traditional centralized cloud requesting for offloading through two types of RAT that compose the basic elements of the HetNet: macrocells and small cells.

Recalling the characteristics the system is premised on, we can summarize:

– a certain application requires O operations and D data,

– the speeds of the mobile device and of the cloud server are, respectively, S_{md} and S_{cs}

– the speed of the mobile device for transferring data to the access point is S_{tr}

– the weight coefficients representing, respectively, the fraction of the computational task and the fraction of the data sent for the offloading are γ and δ, satisfying $0 \leq \gamma, \delta \leq 1$

We recall also that the transmission time is mostly due to the access network transfer, by considering as negligible the transfer time on the backbone network between the access point and the cloud server,

Table 3.1: Values and fraction of computation and transmission

Application	O_k	γ_k	$D_k[b]$	δ_k
1 - Real time road traffic analysis	10^7	0.9	10^5	0.25
2 - Mobile Video and Audio Communication	10^5	0.1	10^7	0.07
3 - Mobile Social Networking	10^7	0.7	10^7	0.35

due to the higher data rate, and the return time from the cloud server to the user terminal because the amount of data in response to the elaboration in the cloud server is small with respect to the data sent toward the cloud server [32, 33].

The values γ and δ, previously found in the optimization procedure of chapter 2 have been considered here for each selected application. We will focus on three application types, characterized by a specific amount of required operations O_k, amount of data that need to be exchanged D_k, fraction of offloaded computational tasks γ_k, and fraction of offloaded data δ_k, as defined in Table 3.1. The considered application classes are:

1. Real time road traffic analysis: the applications aiming to optimize the route toward a certain destination (e.g., navigation applications);
2. Mobile Video and Audio Communications: the applications that elaborates user generated audio and video content;
3. Mobile Social Networking: the applications used for social networking.

In order to describe the throughput, the consumed energy and the time spent by a device for the computation - the three measures we chose for the QoS evaluation - let us focus now on the network infrastructure, by considering a generic couple composed by the i-th access point and the j-th SMD.

Throughput $S_{tr,ij}$ The throughput $S_{tr,ij}$ is affected by the bandwidth BW_i of the i-th access point, by the distance d_{ij} between the access point and the SMD, by the signal to noise ratio SNR_i at the receiver, and by the number of devices n_i already connected to the i-th access point. By resorting to the Shannon formula, the throughput $S_{tr,ij}$ can be written as:

$$S_{tr,ij} = \frac{BW_i}{n_i} \cdot \log_2 \left(1 + \frac{SNR_i}{d_{ij}^2}\right) \tag{3.1}$$

Energy $E_{part\_od,ijk}$ The energy spent for the partial offloading can be written as the sum of the energy spent to perform locally a part of the task and the energy spent during the idle period and during the transmission of the remaining part of the task to the cloud; the idle period corresponds to the amount of time needed by the cloud to perform the computation: we suppose that during this time the SMD remains in an idle state. In this case it is possible to derive the overall spent energy for the k-th application as:

$$E_{part\_od,ijk} = P_{l,j} \times \frac{(1-\gamma_k)\cdot O_k}{S_{md,j}} + P_{id,j} \times \frac{\gamma_k \cdot O_k}{S_{cs}} + P_{tr,ij} \times \frac{\delta_k \cdot D_k}{S_{tr,ij}} \tag{3.2}$$

where $P_{l,j}$ corresponds to the power consumption for performing the local computation by the j-th SMD, $P_{id,j}$ is the power consumption of the j-th SMD in idle state, $P_{tr,ij}$ is the power consumption of the j-th SMD for transmitting the data to the i-th access point, and $S_{md,j}$ is the computing speed in operations per second of the j-th SMD.

Time $T_{part\_od,ijk}$ The computation time for executing the application can be written as the maximum value between the time needed to compute the local portion of the task and the time needed for the offloaded portion; we have supposed that the two phases can be performed at the same time, so that the overall time corresponds to the maximum value:

$$T_{part\_od,ijk} = \max\left(\frac{(1-\gamma_k)\cdot O_k}{S_{md,j}}; \quad \frac{\gamma_k \cdot O_k}{S_{cs}} + \frac{\delta_k \cdot D_k}{S_{tr,ij}}\right) \tag{3.3}$$

3.3 User-Satisfaction Based Utility Function

We introduce, for each of the three quality parameters taken in consideration, a function representing the QoS degree perceived by the user. The functions are modeled as sigmoid curves, since they are well-known functions often used to describe QoS perception [24, 36, 39]. A sigmoid curve can be defined as:

$$U(x) = \frac{1}{1+e^{-\alpha(x-\beta)}} \tag{3.4}$$

where α and β decide the steepness and the center of the curve. The value of α indicates user's sensitivity to the QoS degradation, while β indicates the *acceptable* region of operation. The derivative of the sigmoid function describes the subject perception, so that it does not make sense to give more resources over a certain value above which the derivative of the utility function approximates to zero.

Focusing on the three QoS parameters we are considering, i.e., throughput, energy and time, it is possible to define the related sigmoid functions, by taking into account that the user satisfaction grows with higher throughput values and lower energy and time values. To this aim, concerning the user throughput, it is possible to define the related sigmoid function as:

$$f_1(S_{tr,ij}) = \frac{1}{1+e^{-\alpha_1(S_{tr,ij}-S_{tro,k})}} \tag{3.5}$$

Table 3.2: Reference values for QoS functions

Application	$S_{tro,k}(kb/s)$	$E_{o,k}(W \cdot s)$	$T_{o,k}(s)$
1 - Real time road traffic analysis	0.52	2.9	0.5
2 - Mobile Video and Audio Communication	1.42	3.6	0.1
3 - Mobile Social Networking	0.93	7.1	2

where $S_{tro,k}$ is the objective throughput value for the k-th application.

On the other hand, since the energy and time parameters need to decrease for increasing the user satisfaction, the related cost functions need to be decreasing sigmoid functions defined as:

$$f_2(E_{part\_od,ijk}) = 1 - \frac{1}{1 + e^{-\alpha_2(E_{part\_od,ijk} - E_{o,k})}} \tag{3.6}$$

$$f_3(T_{part\_od,ijk}) = 1 - \frac{1}{1 + e^{-\alpha_3(T_{part\_od,ijk} - T_{o,k})}} \tag{3.7}$$

The parameter α_q ($q = 1, 2, 3$) is defined as the steepness of f_q and is related to the user's sensitivity to the QoS degradation of the q-th parameter. The parameters $S_{tro,k}$, $E_{o,k}$ and $T_{o,k}$ are the center points of the curves f_q, indicating the *acceptable* region of operation. $S_{tro,k}$ and $T_{o,k}$ are reference values for the data transmission rate and the computing time related to the type of application requested, whereas $E_{o,k}$, the energy spent to compute le application locally, is also associated to the type of device in addition to the type of application. For the goal of this study, referring to network analysis rather than SMD's types analysis, we considered values of $E_{o,k}$ dependent only to application types. The values of $S_{tro,k}$, $E_{o,k}$ and $T_{o,k}$ in relation with the application classes are defined in Table 3.2.

On the basis of the QoS sigmoid functions, we introduce a model developed from the economic concept of utility function [39]. By focusing on the access point i (related to the type of RAT) and the SMD j, the cost function for the association of the j-th SMD to the i-th access point is given by:

$$U_{ij} = c_1 \cdot f_1(S_{tr,ij}) + c_2 \cdot f_2(E_{part\_od,ijk}) + c_3 \cdot f_3(T_{part\_od,ijk}) \tag{3.8}$$

where $S_{tr,ij}$, $E_{part\_od,ijk}$ and $T_{part\_od,ijk}$ are the QoS parameters related to the connection between the i-th access point and the j-th SMD for partial offloading of the k-th application. The weight parameters c_q are associated to the importance of the respective quality-related parameters in the performance of the application. For example, for an application of type 1 (real time road traffic analysis) a high weight c_1 (related to S_{tr}) is required, rather then in an application of type 2 (Mobile Video and Audio Communication) where it is more important to have a low delay time, therefore a

Table 3.3: Weight parameters of Utility Function

Application	c_1	c_2	c_3
1 - Real time road traffic analysis	0.6	0.2	0.2
2 - Mobile Video and Audio Communication	0.2	0.2	0.6
3 - Mobile Social Networking	0.2	0.6	0.2

high value of c_3.

The weight parameters c_q are normalized with respect to a certain application, so that it is possible to assume that:

$$\sum_{h=1}^{3} c_h = 1.$$

for each application k. Table 3.3 shows the considered weight parameters for each type of application; it is worth to be noticed that, higher is a certain parameter c_k for a certain application k, higher is the importance of the related QoS parameter for the selected application.

Cell Association

The above defined utility function is at the basis of the CA scheme that allows the selection of the *best* access point by the SMD, for respecting the requirements of the considered applications; whenever a SMD requests to offload an application, the utility function is evaluated for each access point of the network. The SMD will connect to the access point with the maximum utility function.

The selection of a certain access point for establishing the connection could modify the values S_{tr}, E_{part_od} and T_{part_od} for the SMDs already connected with the same access point. Hence, the utility function related to those SMDs is evaluated again, by considering the new incoming SMD. The cell association algorithm is reported in Algorithm 1, where it is possible to note the utility function elaboration and the updating of the utility function for all the SMDs already connected to the selected access point. The CA algorithm is performed for all the SMDs in the scenario.

3.4 Numerical Results

This section deals with the numerical results of the proposed utility function approach by resorting to computer simulations. The smart city scenario has been modeled in Matlab by considering a randomly placed number of SMDs in a deployment area of 1000×1000 m^2, where one LTE eNodeB with channel capacity equal to 100 MHz and three WiFi access points with channel capacities equal

Algorithm 1 Cell Association Algorithm

```
Cell Association Algorithm
for all SMD do
  Cell association request by the $SMD_j$
  for offloading the $App_k$

  for all $RAT_i$ do
    compute $S_{tr,ij}$
    compute $E_{part\_od,ijk}$
    compute $T_{part\_od,ijk}$
    compute $U_{ij}$
    associate $SMD_j$ with $RAT_a$ s.t. $U_{ajk} = \max(U_{ij}) \forall i$
    $RAT_a.n = RAT_a.n + 1$ // update the number of SMDs associated to the $RAT_a$
    for all $SMD_h$ associated to $RAT_a$ do
      compute $S_{tr,ah}$
      compute $E_{part\_od,ahk}$
      compute $T_{part\_od,ahk}$
      compute $U_{ahk}$
    end for
  end for
end for
```

to 22 MHz are positioned to cover the entire area. The SMDs, positioned randomly, will connect to one of the access point/eNodeB, depending on the cell association policy; to this aim we suppose that all the SMDs are capable to connect to both WiFi and LTE. The infrastructures are positioned in the same configuration as represented in Figure 2.3, where the access points are positioned at point (0,0), (500,1000) and (0,1000), and the LTE eNodeB at (500,500).

The SMDs, positioned randomly, request in sequence to offload a random application type and are connected accordingly, on the basis of the presented cell association algorithm. The values S_{md}, P_{id}, P_{tr} and P_l are specific parameters of the mobile devices. We utilized the values of an HP iPAQ PDA with a 400 MHz Intel XScale processor ($S_{md} = 400$) and the following values: $P_l \approx 0.9W$, $P_{id} \approx 0.3W$ and $P_{tr} \approx 1.3W$. As for the cloud server used for the offloading we suppose that $S_{cs} = 8000$ [32].

In the same way used for the utility function, we have resorted to the following values for the steepness of the sigmoidal functions: $\alpha_1 = 1.6 \cdot 10^{-3}$, $\alpha_2 = 10^{-6}$, and $\alpha_3 = 10^{-6}$. These values have been selected after a numerical optimization phase as reported in **??**. We have supposed that the three applications are equally distributed among all the SMDs existing in the environment, so that they have same probability equal to 1/3.

The numerical results are reported by focusing on the performance in terms of average energy consumption, computational time and throughput for each SMD. The numerical results have been compared with three other approaches: local computation, total offloading and nearest node. For the

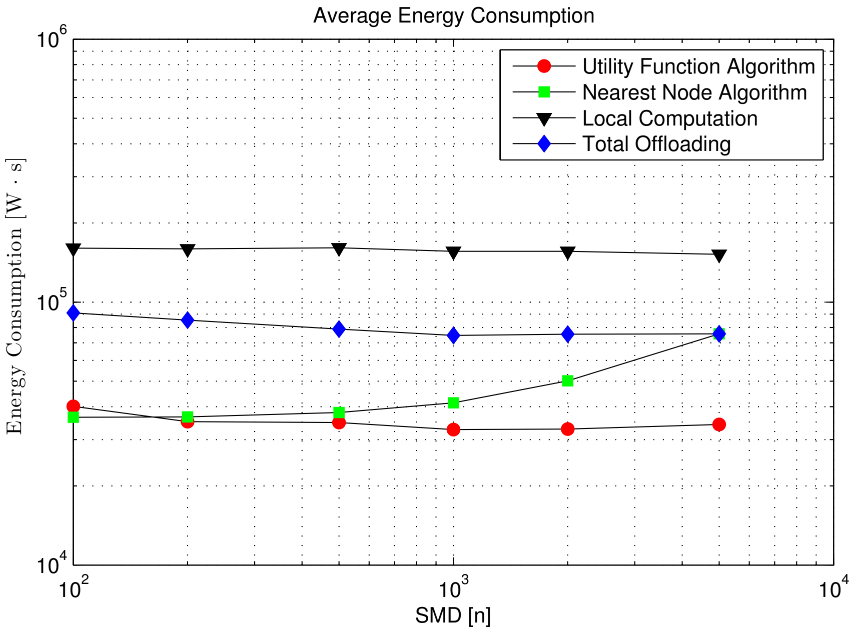


Figure 3.1: Performance results in terms of average energy consumption with a variable number of SMDs.

local computation algorithm is assumed that the computation is performed locally by each SMD; in this case no data is exchanged on the network. In the total offloading algorithm an opposite situation is assumed, since nothing is computed locally, while the entire data load is offloaded to the centralized cloud servers. The nearest node algorithm, finally, assumes that each SMD will connect to the nearest AP/eNodeB.

In Figure 3.1 the results in terms of energy consumption are reported. It is possible to note that both the utility function and the nearest node approaches outperform the local computing and the total offloading. Moreover, it is possible to note that the utility function algorithm allows to have almost the same values for different numbers of nodes, outperforming the nearest node approach for increasing number of SMDs. A similar behavior can be noted in terms of average time for executing the application, in Figure 3.2, where, also in this case, the utility function algorithm outperforms the other approaches.

In Figure 3.3, the performance in terms of average throughput has been reported. In this case the performance for the local computation approach is not reported because in this case no data transfer occurs. It is possible to note that the throughput for the utility function algorithm remains stable, hence giving an optimized value to each SMD, while in the nearest node approach the throughput decreases when the number of SMDs increases.

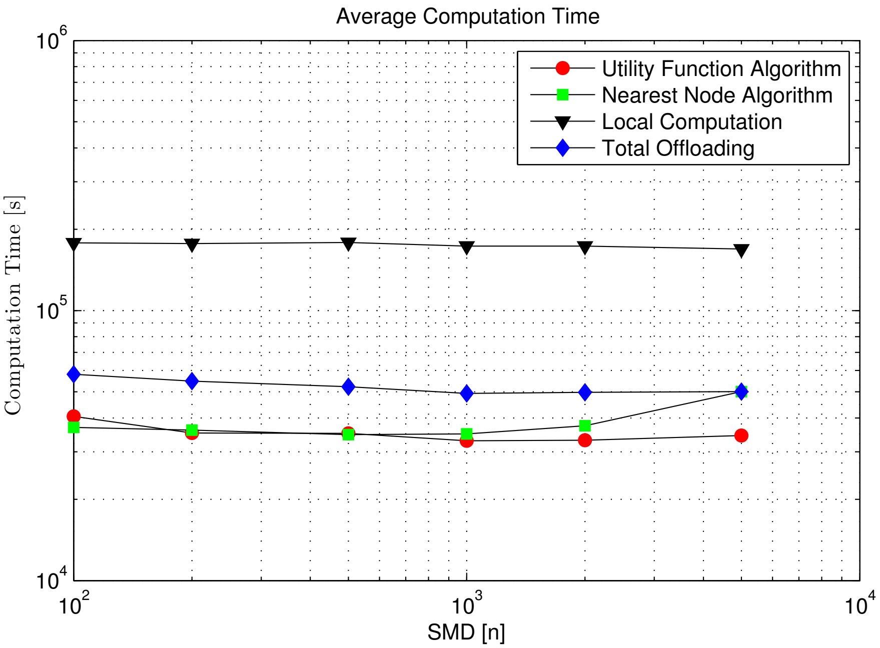


Figure 3.2: Performance results in terms of average computation time with a variable number of SMDs.

3.5 Conclusions

In this chapter we introduced a utility function derived from the economic world aiming to optimize the cell association of the smart devices for achieving low energy consumption and computational time while maximizing the overall throughput. The proposed approach allows to increase the performance with respect to a nearest node association, and with respect to statical approaches where the computation is performed locally or is completely offloaded.

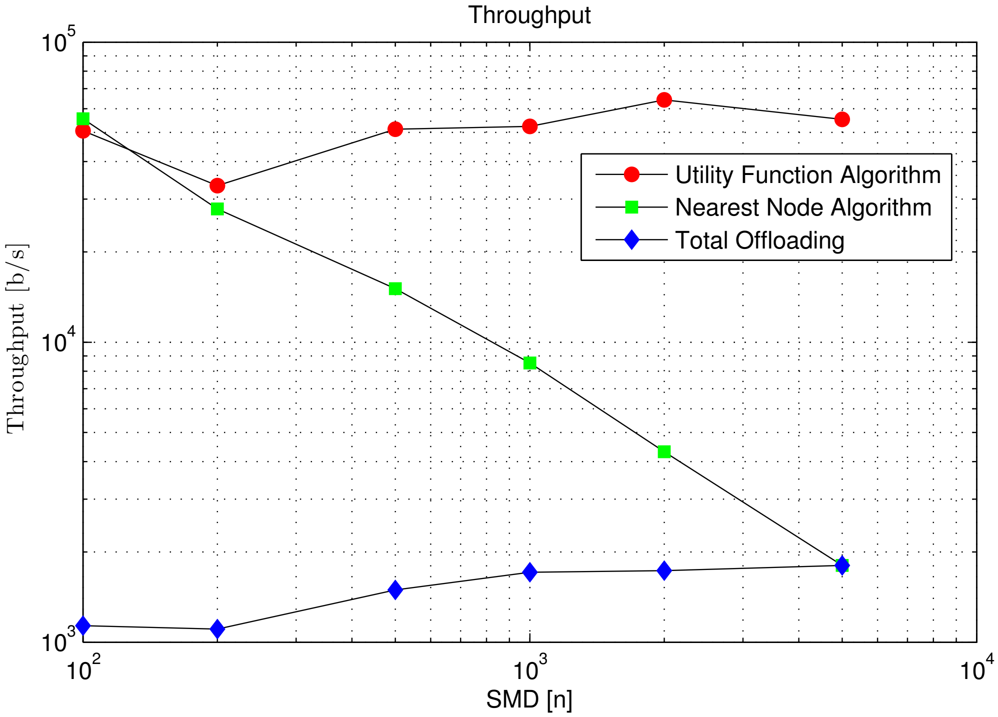


Figure 3.3: Performance results in terms of average throughput with a variable number of SMD.

CHAPTER 4

A Cluster Based Computation Offloading Technique for the UMCC

4.1 Introduction

The evolutionary trends of the Information Society have lead to the definition of the smart city as the most challenging scenario for the IoT applicability. Among several *things* composing a smart city, the computing infrastructures are responsible for giving a distributed elaboration intelligence to the environment aiming at providing unprecedented services and efficiency. The increasing number of devices, appliances, smart-phones and objects connected to the Internet has lead to the introduction of the IoT paradigm, aiming at design the network infrastructure as composed by a moltitude of devices collaborating each others. Among several different functions provided by the IoT devices, the distributed computing approach seems to have an increasing interest allowing to establish an interaction between the IoT and the MCC worlds [7, 40].

This chapter takes into consideration the computation offloading towards other SMDs pooled together, in addition to powerful remote servers, with the aim of an exploitation for shortening execution time and extending battery life. The actual innovation is to take into account different types of mobile cloud infrastructures, considering also a non-trivial extension of the MCC paradigm to the edge of the network, where a collaborative crowd of end-user clients or near-user edge devices share storage, communication, computation, and control. In this context, the issues of mobile device energy saving and performance improvement become increasingly a source of concern, because of the strict constraints on their memory capacity, network bandwidth, CPU speed and battery power [25]. The proposed cluster based computation offloading technique is able to work in a distributed environment, in order to better satisfy the QoS requirements of the SMD users, by minimizing a cost function that takes into account the already mentioned tradeoff between SMD's energy consumption and execution time. By analyzing an actual case of this very complex system, we introduce an algorithm for offloading a real-time navigation application in a distributed fashion, aiming to minimize the execution time.

Although the computation offloading can significantly increase the data processing capabilities for each mobile user, it is challenging to achieve an efficient coordination among the entire set of requesting devices, because this operation can affect the efficiency of the Cloud Computing Infrastructures (CCIs), since the channel capacity has to be shared among all the devices, causing a reduction of the throughput experienced by each user. This could make not convenient the offloading operation [41].

Several excellent works have been done to study cloud and radio resource management issues; among them in the last years there is an increasing importance in the collaborative mobile device offloading [42–44], as well as other considerable studies that face up jointly to clouds and fog architectures [13, 45]. Further, this investigation proposes a system model that takes into account jointly a series of features related to HetNet nodes, different CCIs - in particular a distributed cloud of devices merged together to share resources - and various SMDs characteristics [46, 47].

In this chapter we consider a system where users can interact with all the different topologies of cloud - centralized, cloudlets, and distributed mobile cloud - as described in section 1.2. This allows users to optimize the system performance by offloading a fraction of the application among different cloud structures. The advantage of this approach consists in computing the application that is referred to a specific service exploiting the different cloud infrastructures and network technologies, optimizing important characteristics of the system, i.e. maximize throughput and time computation or minimize the energy used by the SMDs.

A cost function is introduced to optimize the nodes assignment and the resources allocation for distributing the application through the system. In particular we focused on a cluster based solution aiming at exploiting the MCC environment for executing an application in a distributed fashion, for handling a real-time navigation application in a vehicular environment.

4.2 Application Scenario

We consider a UMCC environment where each SMD aims at performing an application, interacting with many CCIs, through different wireless connections, in order to distribute the computation load to one or more CCIs and receive data results from the CCIs themselves. We will consider the possibility of distributing the computation load exploiting multiple CCIs at the same time, for offloading to each one of them a different part of the application. In this chapter we neglect the storage optimization, since each device is supposed to own a storage capacity fitting its computation load. This simplification allows to focus on the proper *computation* offloading optimization.

A SMD can delegate computation functions towards one or more CCIs for performing the requested application. The CCIs can be distinguished on the basis of their features and are classified following the three types described in section 1.2: an ubiquitous centralized cloud infrastructure, many roadside cloudlet infrastructures, and a distributed cloud infrastructure composed by neighboring SMDs pooled together for resource sharing. The centralized cloud refers to an infrastructure with a huge amount of storage space and computing power, virtually infinite, offering the major advantage of the elasticity of resource provisioning. The cloudlets are instead fixed small cloud infrastructures available in a delimited area, installed in proximity of the users and provided with a dedicated transmission node that supports the wireless communication into this area. Finally, the distributed cloud is a Mobile Ad-hoc Network (MANET) composed by the neighboring SMDs.

From the computational point of view, the centralized cloud allows to reduce the computing time by exploiting powerful processing units, but it could suffer from the distribution latency, due to the remote data transfer. Cloudlets, instead, allow to decrease the distribution latency with respect to the centralized cloud, but the storage capacity and the computation power are reduced. Finally, a distributed cloud is composed by the SMDs themselves sharing their own amount of resources depending on the instantaneous usage.

The communication side consists of all the wireless connections available within the selected smart city environment allowing to exploit the centralized cloud, various cloudlet nodes and the SMDs themselves. Aim of the selected system is to divide a certain smart city application into different parts and distribute them among the available nodes. In case a fraction of an application is distributed towards the centralized cloud infrastructure, one of the available HetNet nodes is used for the offloading operation. In case of computation offloading towards a roadside cloudlet infrastructure, the only available node is the one provided in the proximity of the cloudlet itself. In case of computation offloading towards other SMDs belonging to the distributed cloud infrastructure, the transferring operation is made directly by the SMDs. Besides, part of the application can be computed locally by the SMDs requesting the application. Referring to Figure 1.3, we consider the complete distribution, exploiting each of the CCIs represented.

4.3 The Partial Distributed Offloading Model

Even if every SMD of the system can simultaneously requests the computation offloading of one or more applications, we are focusing on a scenario where a single requesting smart mobile device wants to perform an application *App*, whereas all the other SMDs are considered only for receiving and computing it. This simplification does not prevent the generalization of the system, as we can consider the general case as an extension composed by the overlapping of many simplified cases.

The application *App* is defined through the number of operations to be executed, O, and the amount of data to be exchanged, D. The *App* is running on a certain SMD, named Requesting

Smart Mobile Device RSMD, that is the SMD requesting to the CCIs to execute the *App*. We focus for the moment on the presence of a single *App* and a single requesting RSMD.

We suppose that the system demands some features in terms of QoS, in order to have a reliable execution of the *App*. We have taken into account:

- *latency*: the interval between a task of the application is requested and its results are acquired,
- *energy consumption*: the amount of energy the RSMD consumes for performing the application.

Thus, being T_{RSMD} and E_{RSMD}, respectively, the amount of time and energy spent by the RSMD for offloading the application, the system optimization consists in minimizing a weighted sum of the normalized T_{RSMD} and E_{RSMD}, giving more weight to the first or the second parameter depending on the importance of minimizing the *energy consumption* or the *latency*:

$$w_E \frac{E_{RSMD}}{E_o} + w_T \frac{T_{RSMD}}{T_o} \tag{4.1}$$

where w_E and w_T are the weight coefficients, and E_o and T_o reference values.

Both T_{RSMD} and E_{RSMD} depend on the parameters O and D of the *App* and on the overall throughput η_{RSMD} between the nodes and the RSMD, hence, we can rewrite the eq. (4.1), for emphasizing this dependency, as:

$$w_E \frac{E_{RSMD}(O, D, \eta_{RSMD})}{E_o} + w_T \frac{T_{RSMD}(O, D, \eta_{RSMD})}{T_o} \tag{4.2}$$

The system is composed by M available HetNet nodes for offloading towards the centralized cloud (marked as *HN*), N cloudlets nodes (marked as *CL*), and K SMDs' nodes (marked as *MD*), for sharing the computation in the distributed cloud. Thus, being η_{Xi} the throughput offered by the i^{th} node of type X, where X stands for *HN*, *CL*, and *MD*, it results that η_{RSMD} is the sum of the throughput offered by the nodes available for the transfer operation:

$$\eta_{RSMD} = \sum_{i=1}^{M} \eta_{HNi} + \sum_{i=1}^{N} \eta_{CLi} + \sum_{i=1}^{K} \eta_{MDi} \tag{4.3}$$

where we suppose that the RSMD can connect ideally with all the available nodes. The throughput η_{Xi} is related to the number of SMDs n_{Xi} connected to the i^{th} node and the channel capacity BW_{Xi} of the i^{th} node, and can be expressed by resorting to the Shannon Formula:

$$\eta_{Xi} = \frac{BW_{Xi}}{n_{Xi}} \cdot \log_2 \left(1 + \frac{SNR}{{d_{Xi}}^2} \right) \tag{4.4}$$

where *SNR* is a reference signal to noise ratio value of the RSMD at a distance equal to 1 m and d_{Xi} is the distance between the RSMD and the i^{th} node. We have considered for simplicity a no fading-affected channel; this simplification does not affect the operating principles of the framework. Furthermore, we assume a uniform distribution of the available bandwidth among all the n_{Xi} nodes.

T_{RSMD} can be defined as the sum of the time spent to perform locally part of the task, T_l, plus the time spent to transmit data to the clouds, T_{tr}, plus the time spent during the idle period, T_{id}, waiting for the computation performed outside and the results transmitted back. In this case we assume that the transmission time is mostly due to the access network, because the backbone network data rate is much higher. Hence, we can write:

$$T_{RSMD} = T_l + T_{tr} + T_{id} \tag{4.5}$$

T_l can be evaluated as the ratio between the fraction of operation computed locally, $\alpha_0 O$ (where α_0 is the percentage of operations O computed locally), and the computing speed of the RSMD, f_l:

$$T_l = O\frac{\alpha_0}{f_l} \tag{4.6}$$

T_{tr} is the sum of the transferring times towards each node, corresponding to the ratio between the amount of transferred data $\beta_{Xi}D$ (where β_{Xi} is the percentage of transferred data D to the i^{th} node), and the throughput of the node, η_{Xi}. Thus we can write:

$$T_{tr} = D\left(\sum_{i=1}^{M}\frac{\beta_{HNi}}{\eta_{HNi}} + \sum_{i=1}^{N}\frac{\beta_{CLi}}{\eta_{CLi}} + \sum_{i=1}^{K}\frac{\beta_{MDi}}{\eta_{MDi}}\right) \tag{4.7}$$

T_{id} depends on the starting time and the duration of the offloaded computations. Since each CCI is able to compute it own task independently and simultaneously with the other CCIs, and since an offloaded computation cannot begin until all the data needed for performing it are provided, T_{id} would be optimized.

We consider for simplicity the worst case for T_{id}, i.e., the maximum value among every duration of the offloaded computations. Furthermore, by defining as f_{CLi} and f_{MDi} the computing speed of the i^{th} cloudlet and the i^{th} SMD, respectively, and f_{CC} the computing speed of the centralized cloud, and considering α_{Xi} the precentage of computed operations O within the i^{th} node, T_{id} can be written as:

$$T_{id} = O \operatorname*{arg\,max}_{\substack{i=1,\dots,M \\ j=1,\dots,N \\ k=1,\dots,K}} \left\{\frac{\alpha_{HNi}}{f_{CC}}, \frac{\alpha_{CLj}}{f_{CLj}}, \frac{\alpha_{MDk}}{f_{MDk}}\right\} \tag{4.8}$$

E_{RSMD} can be defined as the sum of the energy spent to perform locally part of the task, E_l, plus the energy spent to transmit data to the clouds, E_{tr}, plus the energy spent during the idle period, E_{id}, waiting for the computation performed outside and the results transmitted back:

$$E_{RSMD} = E_l + E_{tr} + E_{id} \tag{4.9}$$

E_l is obtained by multiplying the RSMD's local computation power consumption, P_l, by the time consumed for computing locally the application:

$$E_l = P_l \frac{\alpha_0 O}{f_l} \tag{4.10}$$

E_{tr} is determined by the power consumption of the RSMD for transmitting, P_{tr}, multiplied by the transferring time. Thus we can write:

$$E_{tr} = P_{tr} D \cdot \left(\sum_{i=1}^{M} \frac{\beta_{HNi}}{\eta_{HNi}} + \sum_{i=1}^{N} \frac{\beta_{CLi}}{\eta_{CLi}} + \sum_{i=1}^{K} \frac{\beta_{MDi}}{\eta_{MDi}} \right) \tag{4.11}$$

E_{id} corresponds to the idle time T_{id}, thus, taking into account the same considerations used for T_{id}, we can write the following:

$$E_{id} = P_{id} O \operatorname*{arg\,max}_{\substack{i=1,\dots,M \\ j=1,\dots,N \\ k=1,\dots,K}} \left\{ \frac{\alpha_{HNi}}{f_{CC}}, \frac{\alpha_{CLj}}{f_{CLj}}, \frac{\alpha_{MDk}}{f_{MDk}} \right\} \tag{4.12}$$

Thus, the optimization model consists in finding the values α_{Xi} and β_{Xi} that minimize 4.2, by taking into account the relationships defined in Equation 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, and 4.12. The model constraints are derived from the observation that the sum of the offloaded fractions must be equal to 1, thus the optimization problem becomes:

$$\underset{\alpha_{Xi}, \beta_{Xi}}{\text{minimize}} \left\{ w_E \frac{E_{RSMD}(\alpha_{Xi}, \beta_{Xi})}{E_o} + w_T \frac{T_{RSMD}(\alpha_{Xi}, \beta_{Xi})}{T_o} \right\} \tag{4.13a}$$

$$\text{s.t.} \quad \alpha_0 + \sum_{i=1}^{M} \alpha_{HNi} + \sum_{i=1}^{N} \alpha_{CLi} + \sum_{i=1}^{K} \alpha_{MDi} = 1 \tag{4.13b}$$

$$\sum_{i=1}^{M} \beta_{HNi} + \sum_{i=1}^{N} \beta_{CLi} + \sum_{i=1}^{K} \beta_{MDi} = 1 \tag{4.13c}$$

The obtained minimization problem cannot easy solved, so that, in the following section, we will resort to a sub-optimal solution based on a clusterization approach.

4.4 Cluster based optimization

The idea is to divide the urban area in subareas having range r; each SMD can share resources only with the other SMDs, cloudlets, and HetNet access points placed in the same subarea. This approach, even if sub-optimal, can simplify the problem by reducing the amount of concurrent devices that are involved in the computation offloading. This means, on the other hand, that we have to select the most appropriate cluster size.

If we define with M_r, N_r and K_r, the number of HetNet access points, cloudlets and SMDs, respectively, within the cluster having range r, it is possible to write the transferring time needed by a single RSMD as:

$$T_{tr,r} = D\left(\sum_{i=1}^{M_r}\frac{\beta_{HNi}}{\eta_{HNi}} + \sum_{i=1}^{N_r}\frac{\beta_{CLi}}{\eta_{CLi}} + \sum_{i=1}^{K_r}\frac{\beta_{MDi}}{\eta_{MDi}}\right) \tag{4.14}$$

Let us suppose that among the K_r SMDs within a cluster, $\bar{K}_r \leq K_r$ have an active task to be computed that can be offloaded to the surrounding nodes. Without going into the details of a scheduling algorithm to be used by the cluster based SMDs, we suppose that the *Apps* are computed by all the SMDs that act as a pool of resources, by exploiting a RAN as a Service (RANaaS) cloud model [48]. In this case it is possible to calculate the overall transferring time as:

$$T_{tr,r}^{\text{tot}} = \sum_{j=0}^{\bar{K}_r}\sum_{i=1}^{M_r}\frac{D_j\beta_{HNi}}{\eta_{HNi}} + \sum_{j=0}^{\bar{K}_r}\sum_{i=1}^{N_r}\frac{D_j\beta_{CLi}}{\eta_{CLi}} + \frac{\sum_{j=0}^{\bar{K}_r}\sum_{i=1}^{K_r}D_j\beta_{MDi}}{\sum_{i=1}^{K_r}\eta_{MDi}} \tag{4.15}$$

where D_j is the amount of *App* data requested by the j^{th} RSMD. Similarly, it is possible to write the overall energy needed by the $\bar{K}_r$ nodes for offloading the task as:

$$E_{tr,r}^{\text{tot}} = \sum_{j=0}^{\bar{K}_r}\sum_{i=1}^{M_r}\frac{P_{tr,j}D_j\beta_{HNi}}{\eta_{HNi}} + \sum_{j=0}^{\bar{K}_r}\sum_{i=1}^{N_r}\frac{P_{tr,j}D_j\beta_{CLi}}{\eta_{CLi}} + \frac{\sum_{j=0}^{\bar{K}_r}\sum_{i=1}^{K_r}P_{tr,j}D_j\beta_{MDi}}{\sum_{i=1}^{K_r}\eta_{MDi}} \tag{4.16}$$

From (4.15) and (4.16), it is possible to observe that the transferring time related to the offloading within the SMDs cluster is proportional to the square of the number of the SMDs, i.e., $T_{tr,r}, E_{tr,r} \propto K_r^2$, whereas it is linearly proportional to the number of SMDs in case only the centralized cloud or the cloudlets are used, i.e., $T_{tr,r}, E_{tr,r} \propto K_r$; hence, it is possible to state that there is an optimal number of devices K_r^o within a cluster such that, for $K_r \leq K_r^o$ the offloading towards the centralized cloud is the optimal solution, whereas for $K_r > K_r^o$ the best performance is obtained for the distributed cloud. This means that the cluster size affects the optimization problem.

Since the numbers of SMDs affect primarily the transferring time and energy consumption, we focus our attention on their behavior, without focusing on the idle and local time and energy consumption. Hence, the optimization problem in (4.13), can be rewritten as:

$$\underset{r}{\text{minimize}}\left\{w_E\frac{E_{tr,r}^{\text{tot}}(r)}{E_o} + w_T\frac{T_{tr,r}^{\text{tot}}(r)}{T_o}\right\} \tag{4.17}$$

where the aim is to minimize the linear combination of latency and energy consumption by selecting the cluster size r_o that contains the optimal number of devices K_r^o: this corresponds to estimate the density of the surrounding nodes and select the cluster size by comparing their amount with the optimal number. It is worth to be noticed that the density estimation of a distributed network has been studied in the literature by resorting to specific algorithms, such as in [49]. Hence, for a given scenario, once estimated the SMDs density, it is possible to set the cluster size based on K_r^o.

Once set the cluster size, with respect to the offloading problem, if $K_r \leq K_r^o$, it is convenient to set $\sum_{i=1}^{M_r} \alpha_{HNi} = 1$ and $\sum_{i=1}^{M_r} \beta_{HNi}$=1, that corresponds to a complete offloading towards the centralized cloud, whereas it is better to perform the offloading towards the nodes in the cluster for $K_r > K_r^o$. In this case we consider $\alpha_{CLi} = \beta_{CLi}$ and $\alpha_{MDi} = \beta_{MDi}$ for every node of the distributed cloud, since it is reasonable that the amount of processed data is proportional to the computation load; furthermore, we choose α_{CLi} and α_{MDi} s.t. $\alpha_{CLi}/f_{CLi} = \alpha_{MDi}/f_{MDi}$ for every node, since this is the value that minimize Equation 4.8 and, hence, Equation 4.5.

4.5 Numerical Results

In order to prove the effectiveness of the proposed approach, we resort to a typical scenario for an Intelligent Transportation System (ITS) proposed by Yu et al. [20], composed by the three-layered hierarchical cloud architecture for vehicular networks where the SMDs exploit cloud resources and services in an environment composed by vehicular clouds (i.e., the distributed topology), a set of roadside cloudlets, and a centralized cloud (see Fig. 4.1).

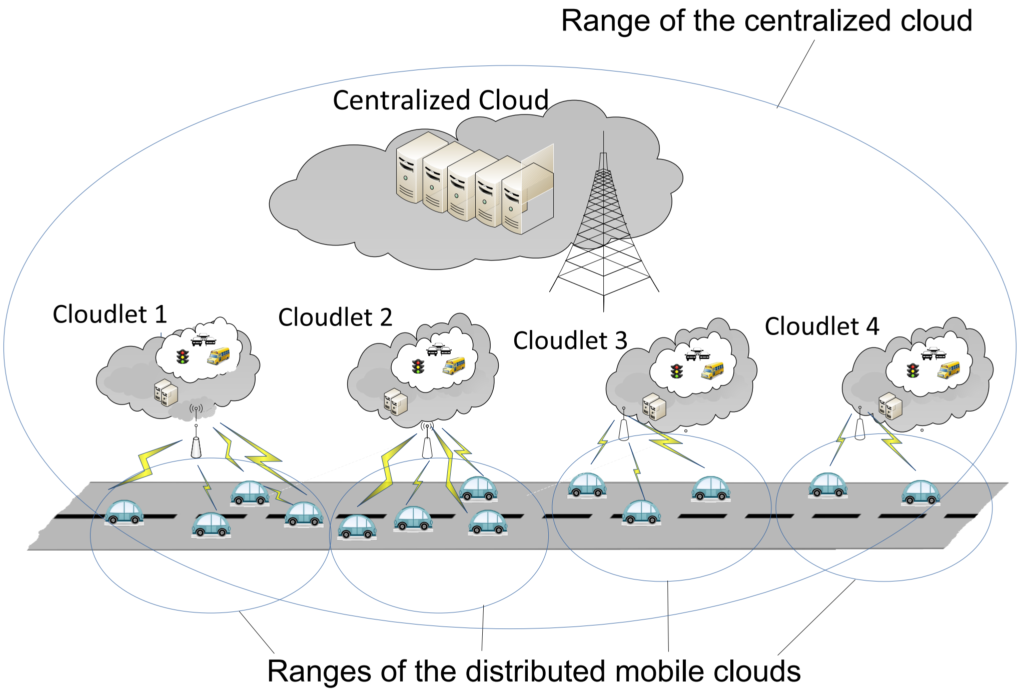

Range of the centralized cloud
Centralized Cloud
Cloudlet 1
Cloudlet 2
Cloudlet 3
Cloudlet 4
Ranges of the distributed mobile clouds

Figure 4.1: Realtime navigation with computation resource sharing, connection with the centralized cloud, the nearest cloudlet and near vehicles

All the devices are supposed to be connected for improving transportation safety, relieve traffic congestion, reduce air pollution and enhance comfort of driving. The selected real-time application, unlike a traditional navigation - which only provides static geographic maps - could be able to offer dynamic three-dimensional maps and adaptively optimize routes based on traffic data mining. Thus, the computational time requirement is of utmost importance. On the other hand, considering that the great part of the involved mobile devices can be recharged directly from the cars in which they are placed, they do not need a restrictive energy consumption requirement. Due to these considerations, we focused the optimization problem for this scenario on a sub-optimal solution minimizing T_{RSMD}, i.e. we consider the weight coefficients $w_E = 0$ and $w_T = 1$ and we assumed that all the devices are requesting the same application. Also the storage requirement could be neglected, while we established that the number of operations O to be executed is equal to 10^6 flops and the amount of exchanged data D by each App is equal to 0.5 Mbit.

In our scenario we considered a linear road 1000 m long with the presence of one centralized cloud with a computation rate f_{CC} equal to 10^3 Gflops, obtained by an Amazon EC2 Cluster GPU instance, through a unique Proxim Tsunami MP 8250 Base Station Unit 4G placed in the middle of the road (at point 500), with a channel capacity equal to 300 Mbps.

A set of four roadside units are the cloudlets evenly positioned at the relative positions of 125, 375, 625, 875 m respectively. Every cloudlet is a compute-box using Nvidia GT520 GPUs with a computing capability f_{cl} equal to 150 Gflops [50]. The offered throughput of the cloudlet η_{cl} is 40 Mbps (considering one LTE cloudlet-devices link), with a coverage range r_{cl} equal to 250 m, so that each cloudlet covers only a limited part of the road.

The SMDs are placed along the route with a uniform distribution, simulating a road where a fluent traffic is present. The SMD computational speed f_{dev} is equal to 10 Gflops, while their offered throughput η_{dev} is equal to 10 Mbps - considering to use the IEEE 802.11p vehicle-to-vehicle standard - and a reference SNR equal to 30 dB. We considered also, for the energy evaluation, the parameters of an HP iPAQ PDA with P_l equal to 0.9 W, P_{id} equal to 0.3 W, P_{tr} equal to 1.3 W.

We have chosen different cases for partitioning the subareas of the distributed clouds, with ranges equal to 250 m, 50 m, 25 m, and 10 m, where every SMD can share resources with the other SMDs placed in the same subarea.

In Figure 4.2 the performance in terms of average latency for an increasing number of SMDs by comparing the total offloading towards the centralized cloud and the distributed clouds is depicted. Four possible clusterizations for the distributed cloud infrastructure are considered; furthermore the total offloading to the nearest cloudlet is also considered. These are compared with the local computation as a benchmark. Moreover, the optimal solution is reported, where the range is selected based on the SMD density.

It is possible to see that by varying the number of SMDs inside the considered area, the most efficient CCIs, to which is convenient to offload, change. In particular for a low density of SMDs

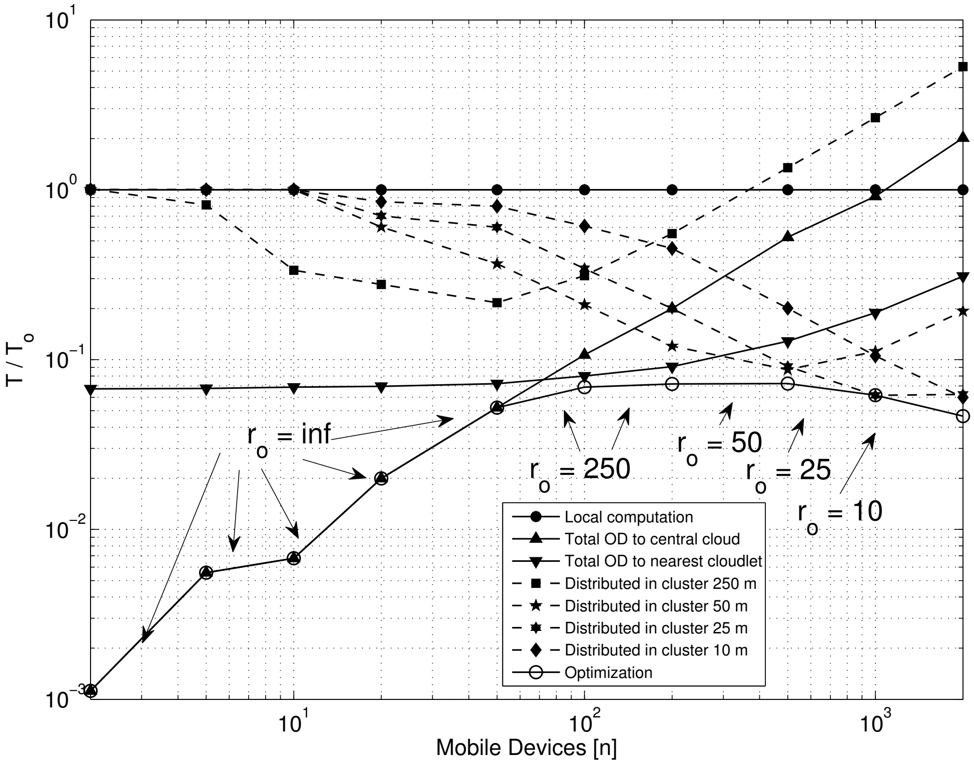


Figure 4.2: Average latency vs the number of RSMDs, in a fluent traffic scenario

it is convenient to offload towards the centralized cloud, corresponding to set the cluster range to infinity, while the cloudlets are convenient just over. This is because in case of low density the SMDs are isolated and, moreover, their communications through the eNodeB -i.e., centralized cloud or cloudlets- have a limited intra-system interference. However, when the number of SMDs increases it is possible to see that the use of the distributed cloud become more convenient, with the cluster size reducing with the increasing of the SMDs. The optimal solution allows to select the best cluster range based on different density situations by approaching always the best cluster size. It is worth to be noticed that the optimal number of SMDs K_r^o, in this case, is equal to 50.

In Figure 4.3 the average energy consumption is depicted by considering a variable number of SMDs, and considering the same cluster sizes as in Figure 4.2. Even if the energy issue does not affect the operations of this particular scenario, where all SMDs are supposed to have unlimited energy since reloaded by the cars' resources, we can consider this issue from an echo-friendly point of view. Comparing the trends of the consumed average energy in the various cases, it is undeniable that the clusterization allows to reduce the consumed energy in case of high traffic scenarios as those present in a smart city environment, and the optimal solution allows to select for different populations the best cluster size, and offloading policy.

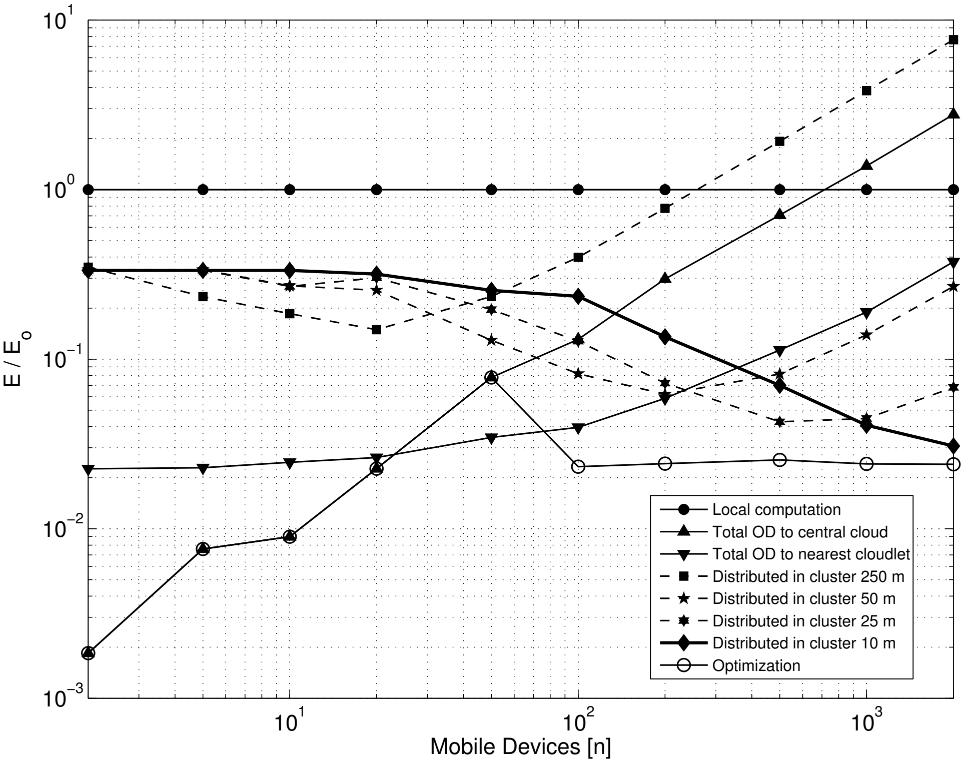


Figure 4.3: Average Energy vs the number of RSMDs, in a fluent traffic scenario

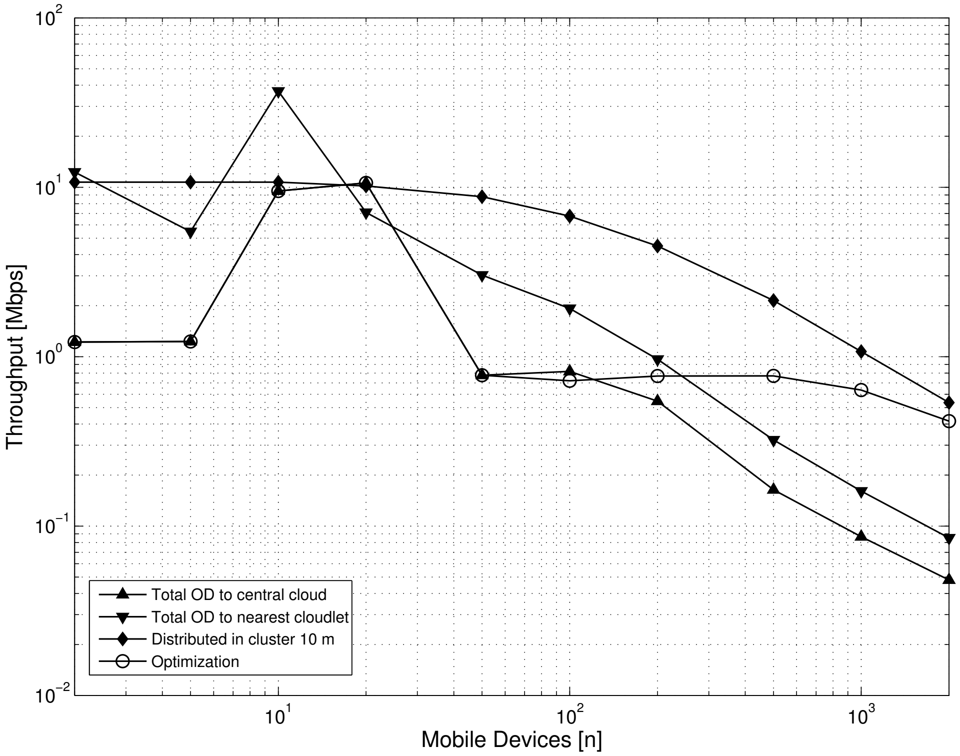


Figure 4.4: Average throughput vs the number of RSMDs, in fluent traffic scenario

Finally, in Figure 4.4 the performance in terms of throughput is shown by comparing the distributed cloud infrastructure with 10 m of range, the total offloading toward the central cloud and the nearest cloudlet and the optimal solution. It is possible to note that the distributed offloading technique outperform the other techniques, except in case of low density of the nodes.

4.6 Conclusions

In this chapter an optimization model for supporting the computation offloading in the UMCC is conceived in a particular IoT scenario, where the objects, consisting of different CCIs, can interact through different types of wireless connections, for creating a distributed computing environment. The distributed computing resource allocation results to be a complex problem due to the presence of several devices and possible wireless connections, thus, we resort to decreasing its complexity by considering a clustering approach for the resource sharing, observing that the use of cluster offloading is useful especially for high density SMD distribution. The numerical results, based on a scenario performing a real-time navigation application of with computation resource sharing, confirm the advantages of the clusterization especially in dense scenarios.

PART III

OPTIMIZATION CONSIDERING A COMMUNITY OF DEVICES

In a typical smart city ecosystem, where a great affluence of mobile devices try to perform applications at the same time, a competition for allocating remote resources occurs, becoming a potential cause of delay and energy consumption. In such a large-scale and dynamic system, relevant optimization challenges concerning the operational assignment of computation offloading become a community issue. Most of these challenges are related to the minimization of the necessary time and energy employed for satisfying the mobile users' requests.

In this part of the thesis the problem of CA in a UMCC framework is analized, considering the system as a community, thinking about the improvement of the collective performance and not only the one of a single device. In chapter 5 three different methods for solving the offloading-related CA problem are compared: the first leads to the optimal solution, but it requires the complete information about the final position of the requesting devices and, therefore, cannot be used in a realistic scenario with online demands. The other two get sub-optimal solutions, but allow connecting each device on the fly –as new requests appear. The first of these two online algorithms is based on a greedy heuristic which chooses 'the best' network node for each single device –selfish behavior. The second one is a probabilistic algorithm that makes use of biased-randomization techniques [51] in order to enhance the quality of the solution from a *social* or collective point of view.

The content of the following chapter was extracted from publications [P2] and [P4].

CHAPTER 5

A Social-Aware Biased-Randomized Algorithm for UMCC

5.1 Introduction

In this chapter the UMCC is exploited for accomplishing applications in a cooperative way while satisfying the QoS requirements for the citizens involved. By taking into account all users in the assignment process, the proposed biased-randomized algorithm allows to enhance social-aware performance as compared with the greedy approach while, at the same time, it allows for online allocation of resources.

Although the computation offloading can significantly increase data processing capability for the single mobile users, it is challenging to achieve an efficient coordination among the entire set of requesting devices when the environment is particularly crowded. In fact, the computation offloading entails data transmission among the devices and the cloud, to make the operation possible. If a great number of users utilize the same wireless resource to delegate computation to the cloud, this operation can affect the efficiency of the access node, since its channel capacity has to be shared among all the devices, causing a reduction of the throughput experienced by each user. This can lead to a crucial use of energy and time spent for data transmission, making not convenient the offloading operation [41].

In this paper we present a probabilistic method that makes use of biased-randomization techniques to solve the CA problem, i.e. to select, among the available access nodes, the one that increases the system efficiency. This method is compared with the global optimization solution, that is shown to be unacceptable from a point of view of time computation, and with a greedy algorithm [52], that instead implement a selfish behavior in allocating the resources to the users. All the three techniques –the probabilistic, the greedy and the optimal– are based on the definition of a proper cost function that takes into account the different requirements and characteristics of the considered UMCC framework. We will consider, for our example and calculation, the throughput, the energy consumption, and the delay as those parameters that drives the optimal allocation of the resources to be offloaded.

The novel proposed algorithm improves the solutions and is easy to implement in real time, thus outperforming by far the solutions provided by the heuristic approach. The underlying biased randomization techniques [51] introduces some degree of randomness using a skewed probability distribution and reaching a solution nearer to the optimal without the need of knowing all the SMDs' positions.

As it will be illustrated in the numerical results, the use of the biased-randomized approach allows to easily enhance the quality of the solutions generated by the original heuristic in different dimensions when considering social or collective performance. It is important to notice that if a uniform probability distribution would be used instead of a skewed one, this improvement would very rarely occur since the logic behind the constructive heuristic would be destroyed and, accordingly, the process would be random but not correctly oriented.

In the context of combinatorial optimization problems, constructive heuristics use an iterative process in order to construct a 'good' and feasible solution. Examples of these heuristics are the savings procedure for the Vehicle Routing Problem [53], the NEH procedure for the Flow-Shop Problem [54], or the Path Scanning procedure for the Arc Routing Problem [55]. In all these heuristics, a 'priority' list of potential movements is traversed during the iterative process. At each iteration, the next constructive movement is selected from this list, which is sorted according to some criteria. The criteria employed to sort the list depends upon the specific optimization problem being considered. Therefore, a constructive heuristic is nothing more than an iterative greedy procedure, which constructs a feasible 'good' solution to the problem at hand by selecting, at each iteration, the 'best' option from a list, sorted according to some logical criterion. Notice that this is a deterministic process, since once the criterion has been defined, it provides a unique order for the list of potential movements. Of course, if we randomize the order in which the elements of the list are selected, then a different output is likely to occur each time the entire procedure is executed. However, a uniform randomization of that list will basically destroy the logic behind the greedy behavior of the heuristic and, therefore, the output of the randomized algorithm is unlikely to provide a good solution.

5.2 Problem Description

The environment consists of an urban area with a pervasive wireless coverage, where several mobile devices are interacting with a centralized cloud infrastructure and request for services from a remote data center. In order to connect the SMDs to the cloud, the presence of different types of RATs that compose the basic elements of the HetNet has been considered. The SMDs can connect to the

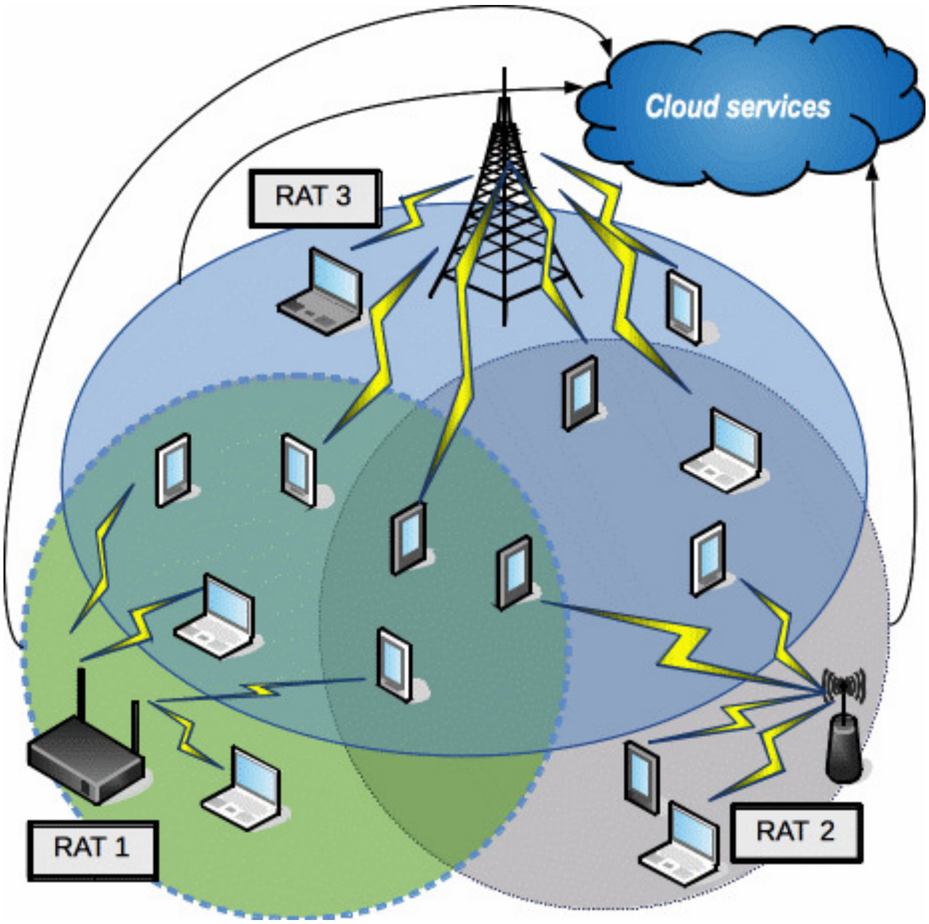


Figure 5.1: The reference scenario with different node types in HetNet used by SMDs for offloading applications to UMCC

cloud, as shown in Figure 5.1, by using the reachable RATs. The selection will depend upon the distance to the RATs as well as on other features that will be detailed below.

The existence of different RATs allows cloud computing systems to complement the resource-constrained mobile devices. In effect, storage and computing tasks can be completed in the cloud, where the appropriate technology is selected among a variety of options. Different strategies can be carried out in order to enhance the network capacity, such as: deployment of relays, distributed antennas, and small cellular base stations (e.g. macrocells, picocells, femtocells), etc. These deployments have occurred indoors, in residential homes and offices, as well as outdoors, in amusement parks and busy intersections.

These new network deployments are typically comprised by a mix of low-power nodes underlying the conventional homogeneous macrocell network. By deploying additional small cells within the local-area range and bringing the network closer to users, they can significantly boost the overall network capacity through a better spatial resource reuse. Inspired by the attractive features and the potential advantages of HetNets, their development have gained much momentum in the wireless industry and research communities during the past few years.

The goal of the system and of every SMDs, indeed, is to choose and exploit effectively the different available RATs for transmitting the application data and for offloading towards the cloud. However, if this operation is not convenient, the application can be computed locally by the requesting SMD. The analized approaches to solve a collective CA problem are based on a cost function related to the number and the localization of the SMDs. The first approach is the optimization of the cost function, considering a relation among the different measures involved, but it can only be used once the position and requests of all the SMDs is known –i.e., it cannot be used for online allocation of resources. Thus, this approach assumes that all users' requests are processed at the same time –or, alternatively, that the allocation decision is postponed until all requests have been received–, which in practice might require customers to wait for their requests to be served.

The second approach adopts a heuristic method that makes use of a 'greedy' behavior consisting on the selection of the 'best next step' from a list of potential constructive movements [52], allowing to find a good solution *on the fly*, so that every SMD can connect to the proper cell without waiting for all the other SMDs taking place in the environment.

The third, finally, is a novel algorithm that makes use of biased-randomization techniques [51] to improve the results obtained with the second.

In the heuristic described for the second algorithm, each agent tries to make the choice that maximizes his/her individual utility function. However, this might lead to sub-optimal solutions from a collective or social point of view, since individual decisions do not take into account a global perspective.

In this context, the main idea behind our third approach is to introduce a slight modification in the greedy constructive behavior. This is done so that the constructive process is still based on the heuristic logic but, at the same time, some degree of randomness is introduced. This random effect is generated throughout the use of a skewed probability distribution: at each step of the constructive process, each potential movement receives a probability of being selected. This measure is higher as more 'promising' is the movement.

In the following of this section the various components of the UMCC described in chapter 1 are recalled emphasizing the dependency from the parameters that play a role in the used algorithms.

Centralized Cloud

The cloud entity C_{cc} is characterized by its own computation speed, f_{cc}. The storage availability of C_{cc} can be considered infinite, and thus it does not represent a constraint in the interaction with the SMDs. Hence, for the centralized cloud it is possible to write:

$$C_{cc} = C_{cc}(f_{cc}) \tag{5.1}$$

HetNet nodes

The wireless HetNet infrastructure is composed of some RAT nodes, each of them characterized by different features:

- Channel Capacity BW: the nominal bandwidth of a certain communication technology that is available for the requesting connecting devices;
- n: the number of devices connected to the RAT;
- $pos_{RAT}(x, y)$: the position in which the RAT is located.

Hence, for the a generic RAT it is possible to write:

$$RAT = RAT(BW, n, pos_{RAT}(x, y)) \tag{5.2}$$

Applications

Even if every SMD of the system can simultaneously request the computation offloading of one or more applications, we focus on a scenario where each SMD demand is restricted to a single application, which is characterized by the number of operations to be executed, O, and the amount of data to be exchanged, D. This simplification does not prevent the generalization of the system, since the general case in which a user requests several applications at a given time can be seen as an aggregation of several single-application requests. Hence, it is possible to express a generic application App as:

$$App = App(O, D) \tag{5.3}$$

SMDs

Each SMD is characterized by the following features which influence the performance of the computation offloading:

- P_{tr}: the power consumed by the SMD to transmit data to the node
- P_l: the power consumed by the SMD for local computation
- P_{id}: the power consumed by the SMD in waiting mode while the computation in performed in the cloud
- f_{md}: the speed to perform the computation locally

- *SNR*: the SNR of the device

- $pos_{md}(x, y)$: the position in which the SMD is located.

Hence, for a generic smart mobile device *SMD*:

$$SMD = SMD(P_{tr}, P_l, P_{id}, f_{md}, SNR, pos_{md}(x, y)) \tag{5.4}$$

Given these entities, with the respective features, and defining $d_{i,j}$ as the distance between the i-th RAT to the j-th SMD, we can resort the Shannon formula for the throughput $S_{i,j}$ delivered by the i-th RAT to the j-th SMD, where k is an attenuation coefficient for the signal propagation:

$$d_{i,j} = |pos_{RAT,i}(x, y) - pos_{md,j}(x, y)| \tag{5.5}$$

$$S_{i,j} = \frac{BW_i}{n_i} \log_2 \left\{1 + SNR_j e^{-kd_{i,j}}\right\} \tag{5.6}$$

In case of computation offloading, Equation 5.6 affects both the measures we are considering for evaluating the QoS:

- the energy consumed by the j-th SMD, i.e.: the sum of the energy spent in transmitting the application data plus the energy consumed while the application is computed on the cloud server;

- the time required to accomplish the application, i.e.: the sum of the time spent in transmitting the application data plus the time for computation on the cloud server.

Considering an overall number of N devices and a total number of M HetNet nodes, the energy and the time spent by the j-th SMD for offloading towards the i-th RAT are, respectively:

$$E_{i,j} = \frac{P_{tr\ i} D}{S_{i,j}} + \frac{P_{id\,i} O}{f_{cc}} \quad i = 1, 2, \ldots, M; j = 1, 2, \ldots, N; \tag{5.7}$$

$$T_{i,j} = \frac{D}{S_{i,j}} + \frac{O}{f_{cc}} \quad i = 1, 2, \ldots, M; j = 1, 2, \ldots, N; \tag{5.8}$$

On the other hand, in case of local computation executed by the j-th SMD, the energy and the time consumed by the SMD are, respectively:

$$E_{0,j} = \frac{P_{l\,j} O}{f_{md\,j}} \quad j = 1, 2, \ldots, N; \tag{5.9}$$

$$T_{0,j} = \frac{O}{f_{md\,j}} \quad j = 1, 2, \ldots, N; \tag{5.10}$$

where the 0 index represents a fictitious node related to this operation.

The SMD requesting to compute an application *App* evaluates if the computation offloading is convenient and which is the RAT to be used. The decision aims at minimizing the energy consumption of the user requesting the service as well as the execution time required to accomplish the application. Therefore, the decision is 'selfish' in the sense that it does not take into account the current and future needs of other users.

We consider, instead, to optimize the global energy consumption, assuming an environment where the devices of the set $\mathcal{N} = \{1, 2, \cdots, j, \cdots, N\}$ are requesting to offload an application using the nodes of the set $\mathcal{M} = \{0, 1, 2, \cdots, i, \cdots, M\}$, where the 0-th element represents the fictitious node related to the local computation.

Defining $x_{i,j}$ as a binary variable that indicates whether SMD$_j$ is assigned to RAT$_i$ or not, and y_j as another binary variable to account for SMD$_j$ computing locally its application, it is possible to write the energy consumed by SMD$_j$ when offloading towards RAT$_i$ (Equation 5.7) by exploiting Equation 5.6

$$E_{i,j} = x_{i,j} \left(\frac{P_{tr\,j}\, D}{\frac{BW_i}{n_i} log_2 \left\{1 + SNR_j e^{-kd_{i,j}}\right\}} + \frac{P_{id\,j}\, O}{f_{cc}} \right) + y_j \left(\frac{P_{l\,j}\, O}{f_{md\,j}} \right) \tag{5.11}$$

$$x_{i,j} = \begin{cases} 1 & \text{if SMD}_j \text{ is assigned to RAT}_i, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if SMD}_j \text{ computes } \textit{App} \text{ locally,} \\ 0 & \text{otherwise.} \end{cases}$$

where n_i is the number of SMDs connected to RAT$_i$, i.e. $n_i = \sum_{l \in \mathcal{N}} x_{i,l}$.

Rearranging the constant terms and the variable terms it is possible to write:

$$E_{i,j} = x_{i,j} \left\{ K_{i,j}^{tr} \sum_{l \in \mathcal{N}} x_{i,l} + K_j^{id} \right\} + y_j E_{0,j} \tag{5.12}$$

The constant part for offloading due to the transmission is computed as $K_{i,j}^{tr} = \frac{P_{tr\,j}\, D}{BW_i log_2 \left\{1 + SNR_j e^{-kd_{i,j}}\right\}}$ plus the constant part $K_j^{id} = \frac{P_{id\,j}\, O}{f_{cc}}$. Therefore the optimization model can be expressed as:

$$\min_{x,y} \; Z = \sum_{\substack{i \in \mathcal{M} \\ j \in \mathcal{N}}} x_{i,j} \left\{ K_{i,j}^{tr} \sum_{l \in \mathcal{N}} x_{i,l} + K_j^{id} \right\} + \sum_{j \in \mathcal{N}} y_j E_{0,j} \tag{5.13a}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{M}} x_{i,j} + y_j = 1 \qquad j \in \mathcal{N} \tag{5.13b}$$

$$x_{i,j}, y_j \in \{0, 1\} \qquad i \in \mathcal{M} \qquad j \in \mathcal{N}; \tag{5.13c}$$

The objective function (Equation 5.13a) minimizes the overall energy spent, while ensuring that all devices are either assigned to a RAT or leaving the computations to be done locally (Equation 5.13b).

5.3 Optimal and Heuristic Solutions

In this section different methods to solve the optimization model are presented. A first approach is to use an off-the-shelf commercial solver, but for real large-size instances it requires unsuitable computation times. Moreover, since we take into account mobile SMDs, thus their position changes in time, the cell association has to be computed often. Therefore, given that the assignation of the SMDs to the RAT nodes has to be done in a very short time, we analyze several heuristic procedures. We focus on a novel Algorithm 4 based on biased randomization strategy, that is proposed to improve the quality of the solution from a social or collective point of view, enhancing a previously proposed greedy Algorithm 3. To evaluate their effectiveness we use the optimal solution presented in subsection 5.3.1, that can only be applied *ex-post*, once all the SMD requests have been revealed. This optimal solution is found using the procedure of Algorithm 2. In particular, since the global optimization needs to know the position and the requests of all the SMDs, it cannot be used for online allocation of resources. Thus, this approach assumes that all users' requests are processed at the same time –or, alternatively, that the allocation decision is postponed until all requests have been received–, which in practice might require customers to wait for their requests to be served.

On the other hand, the heuristic method making use of a 'greedy' behavior, consisting on the selection of the 'best next step' from a list of potential constructive movements [52], allows to find a good solution in real time, so that every SMD can connect to the proper cell without waiting for all the other SMDs taking place in the environment. But with this heuristic, each agent tries to make the choice that maximizes his/her individual utility function, so that this might lead to sub-optimal solutions from a collective or social point of view, since individual decisions do not take into account a global perspective.

5.3.1 Optimal Solution

The previous objective function shown in Equation 5.13 can be seen as an application of the Quadratic Semi-Assignment Problem (QSAP) which is NP-hard [56]. A first approach to find the solution is to use the global optimizer Baron [57]. The Branch-And-Reduce Optimization Navigator derives its name from its combining constraint propagation, interval analysis, and duality in its reduce arsenal with advanced branch-and-bound optimization concepts.

Note that the QSAP not only needs to know the position of all SMDs requests before assigning them to an antenna, but also it takes a time longer than the needed to serve quickly the SMDs, as shown in section 5.4. Therefore, the optimal resolver can be taken only as reference, to compare fast heuristics (which can do the assignation dynamically or with a *wait-and-go*) with an optimal, ideal solution. Algorithm 2 shows the method to find the optimal ex-post solution.

Algorithm 2 Optimal Solution

```
Inputs:
  $\mathcal{M} = \{1, 2, \cdots, M\}$ // $M$ number of RAT nodes
  $\mathcal{N} = \{1, 2, \cdots, N\}$ // $N$ number of SMDs
  $RAT_i = RAT(BW_i, n_i, pos_{RAT,i}(x, y))\ i \in \mathcal{M}$
  $SMD_j = SMD(P_{tr,j}, P_{l,j}, P_{id,j}, f_{md,j}, SNRj, pos_{md,j}(x, y))\ j \in \mathcal{N}$
  $App = App(O, D)$
Output:
  $A$ // assignment vector
  Initialization :
  Compute constants $K^{tr}_{i,j}$ and $K^{id}_j$
  Solve model (??) $\rightarrow x^*_{i,j}, y^*_j$
  for $j = 1$ to $N$ do
    if $y^*_j == 1$ then
      $A \leftarrow A_0$
    else
      for $i = 1$ to $M$ do
        if $(x^*_{i,j} == 1)$ then
          $A \leftarrow A_i$
        end if
      end for
    end if
  end for
```

5.3.2 The Greedy Heuristic

In this sub-section, we recall a heuristic algorithm used to solve the CA problem following a greedy behavior. It has been used also in chapter 3, to resolve the CA problem in the case discussed above, considering a user-satisfaction based utility function. If the offloading operation is advantageous with respect to the local computation, the CA scheme leads to select the 'best' next node from the list of the available ones. The requests of the SMDs appear in time sequence, and the cost function is evaluated on the basis of the current situation, i.e., considering only the previously appeared SMDs. If the offloading cost is less than the cost for the local computation, the SMD will connect to the node which minimizes the cost function, otherwise it will compute the application without connecting. The selection of the i-th node for connecting the j-th SMD modifies the values of the throughput $S_{i,k}$ and consequently of the energy $E_{i,k}$ for the SMDs already connected with the same node, i.e., for $k = 1, 2, \ldots, j-1$. Thus, this strategy, reported in Algorithm 3, does not take into account any forecast of future connections, leading to a sub-optimization due to the randomness of the requests.

Algorithm 3 Greedy Cell Association Heuristic

```
Inputs:
  $\mathcal{M} = \{0, 1, 2, \cdots, M\}$ // $M$ number of RAT nodes
  $\mathcal{N} = \{1, 2, \cdots, N\}$ // $N$ number of SMDs
  $RAT_i = RAT(BW_i, n_i, pos_{RAT,i}(c_x, c_y))\ i \in \mathcal{M} \setminus \{0\}$
  $App = App(O, D)$
Output:
  $X = (x_{i,j})$
  $Y = (y_j)$
  Initialization :
  $RAT_i.n \leftarrow 0\ \forall i \in \mathcal{M} \setminus \{0\}$
  $x_{i,j} \leftarrow 0\ \forall i \in \mathcal{M}\ \forall j \in \mathcal{N}$
  $y_j \leftarrow 0\ \forall j \in \mathcal{N}$
  for $j = 1$ to $N$ do
    $SMD_j = SMD(P_{tr,j}, P_{l,j}, P_{id,j}, f_{md,j}, SNR_j, pos_{md,j}(c_x, c_y))$
    $RAT_i.n \leftarrow RAT_i.n + 1\ \forall i \in \mathcal{M} \setminus \{0\}$
    $calculate\ E_{i,j}\ \forall i \in \mathcal{M}$
    $choose\ x_{i,j}\ \ y_j\ s.t.\ E_{i,j} = min\{\ E_{l,j};\ l \in \mathcal{M}\}$
    $RAT_l.n \leftarrow RAT_l.n - 1\ \forall l \in \mathcal{M}\ s.t.\ x_{l,j} \neq 1$
    $X \leftarrow\ x_{i,j}$
    $Y \leftarrow\ y_j$
  end for
```

5.3.3 Biased-randomized Algorithm

Making use of biased-randomization techniques [51], the proposed probabilistic algorithm is able to find near-optimal solutions in real time, thus outperforming by far the solutions provided by the heuristic approach. The main idea behind our novel approach is to introduce a slight modification in the greedy constructive behavior. This is done in such a way that the constructive process is still based on the heuristic logic but, at the same time, a certain degree of randomness is introduced. This random effect is generated throughout the use of a skewed probability distribution: at each step of the constructive process, each potential movement receives a probability of being selected, being this probability higher for the more *promising* movements.

As it will be illustrated in the numerical results, the use of the biased-randomized approach allows to easily enhance the quality of the solutions generated by the original heuristic in different dimensions when considering social or collective performance, reaching a solution nearer to the optimal, without the need to know all the SMDs' positions.

Also, it is important to notice that if a uniform probability distribution would be used instead of a skewed one, this improvement would very rarely occur since the logic behind the constructive heuristic would be destroyed and, accordingly, the process would be random but not correctly oriented.

To avoid losing the logic behind the heuristic, GRASP meta-heuristics [58] proposes to consider

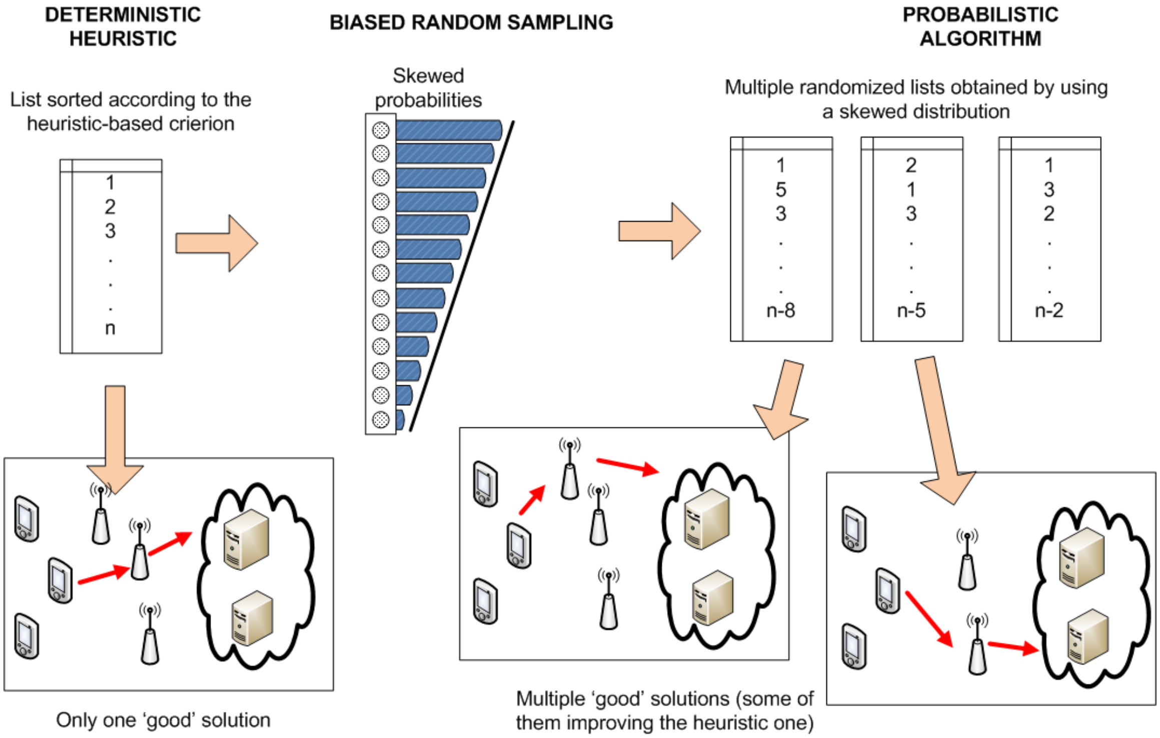

DETERMINISTIC HEURISTIC
List sorted according to the heuristic-based crierion
1
2
3
.
.
.
n
BIASED RANDOM SAMPLING
Skewed probabilities
PROBABILISTIC ALGORITHM
Multiple randomized lists obtained by using a skewed distribution
1 5 3 . . . n-8
2 1 3 . . . n-5
1 3 2 . . . n-2
Only one 'good' solution
Multiple 'good' solutions (some of them improving the heuristic one)

Figure 5.2: Scheme of the biased-randomization approach.

a restricted list of candidates –i.e., a sub-list including just some of the most promising movements, that is, the ones at the top of the list–, and then apply a uniform randomization in the order the elements of that restricted list are selected. This way, a deterministic procedure is transformed into a randomized algorithm –which can be encapsulated into a multi-start process–, while most of the logic or common sense behind the original heuristic is still respected.

The proposed biased-randomization approach goes one step further, and instead of restricting the list of candidates, it assigns different probabilities of being selected to each potential movement in the sorted list. In this way, the elements at the top of the list receive more probabilities of being selected than those at the bottom of the list, but potentially all elements could be selected. Notice that by doing so, we are not only avoiding the issue of selecting the proper size of the restricted list, but we also guarantee that the probabilities of being selected are always proportional to the position of each element in the list. As a result, each time the randomized algorithm is executed, a new probabilistic solution is obtained (Figure 5.2). Some of these solutions will improve the original one provided by the base heuristic and, moreover, the proposed approach allows to offer alternative solutions to choose from, each of them with different properties.

Focusing on our specific problem, the biased-randomization algorithm for the cell association consists of a skewed criteria based on the same cost function utilized for the greedy heuristic, i.e., the SMD's energy consumption for computing the application. When a SMD request occurs, the cost function $E_{i,j}$ is evaluated for every possible RAT node (considering also the fictitious node 0 related to the local computation).

The values are sorted in a way so that the probabilities of being selected depend not only on the cost function $E_{i,j}$, but also on the distances $d_{i,j}$ of the device from the RAT nodes and on the number of devices n_i that potentially can be connected to each RAT node.

In fact, on the basis of the greedy algorithm, a SMD far from all the RAT nodes, but occurred before any other, is associated to a RAT because the cost function value of the offloading is less than the cost function for the local computation. Instead, due to the high probability that another SMD closer to this RAT occurs, the local computation could be a better choice, giving place to the connection of the second closer SMD. To improve the cell association, the criteria for sorting the list of choices takes into account the product $d_{i,j} \cdot E_{i,j}$, with $i \in \{0\} \cup \mathcal{M}$, instead of $E_{i,j}$. The value of $d_{0,j}$ is adjusted empirically and depends on the overall number of SMDs which are expected to interact in the system. We selected a value $d_{0,j}$ depending on the overall number of SMDs which are supposed to interact in the system. The value of $d_{0,j}$ has been evaluated observing the optimal solution given by 5.3.1 in the post-analysis, i.e., when all the SMD positions are already known.

This strategy is reported in Algorithm Algorithm 4.

Algorithm 4 Biased-randomization Algorithm

```
Inputs:
  $\mathcal{M} = \{0, 1, 2, \cdots, M\}$ // $M$ number of RAT nodes
  $\mathcal{N} = \{1, 2, \cdots, N\}$ // $N$ number of SMDs
  $RAT_i = RAT(BW_i, n_i, pos_{RAT,i}(c_x, c_y))\ i \in \mathcal{M} \setminus \{0\}$
  $App = App(O, D)$
Output:
  $X = (x_{i,j})$
  $Y = (y_j)$
  Initialization :
  $d_{0,j} \leftarrow d_0$
  $RAT_i.n \leftarrow n_i\ \forall i \in \mathcal{M} \setminus \{0\}$
  $x_{i,j} \leftarrow 0\ \forall i \in \mathcal{M}\ \forall j \in \mathcal{N}$
  $y_j \leftarrow 0\ \forall j \in \mathcal{N}$
  for $j = 1$ to $N$ do
    $SMD_j = SMD(P_{tr,j}, P_{l,j}, P_{id,j}, f_{md,j}, SNRj, pos_{md,j}(c_x, c_y))$
    $calculate\ d_{i,j}\ ; E_{i,j}\ \forall i \in \mathcal{M}$
    $calculate\ E_{0,j}$ // energy for local computation
    sort $d_{i,j} * E_{i,j}\ ascending\ \forall i \in \mathcal{M}$
    choose $x_{i,j}\ y_j\ s.t.\ E_{a,j} = geoinv(\ E_{i,j};\ i \in \mathcal{M})$
    $X \leftarrow\ x_{i,j}$
    $Y \leftarrow\ y_j$
  end for
```

5.4 Numerical Results

We considered a deployment area similar to that of the previous chapters, i.e. an extension of $1000m^2$, where one LTE eNodeB with channel capacity equal to 100 MHz is positioned at point (500, 500) and three WiFi access points with channel capacities equal to 22 MHz are positioned at point (0, 0), (500, 999) and (1000, 0). Each of them is supposed to cover the entire area and we considered an attenuation coefficient for the propagation k equal to 10^{-3}. Furthermore, similarly to [52], the capacity constraints on the antennas is not considered, assuming there is no limitation on the number of customers.

The SMDs are placed in the deployment area according to a uniform distribution. In particular, we have chosen a controlled random number generation of type Mersenne Twister with seed equal to 1. The values of f_{md}, P_{id}, P_{tr}, P_l, and SNR are specific parameters of the mobile devices. We considered the values of an HP iPAQ PDA with a 400 MHz Intel XScale processor ($f_{md} = 400$ MHz) and the following values: $P_l = 0.9W$, $P_{id} = 0.3W$, $P_{tr} = 1.3W$, and $SNR = 1000$. Regarding the cloud server, we suppose a computation speed $f_{cs} = 10^6$ MHz [32]. We have chosen an application which is accomplished through a number of operation $O = 10^7$ and, if offloaded, needs a data transfer $D = 10^4$ bits.

We have compared the cell association configurations resulting from the greedy heuristic, the biased-randomization algorithm, and the Baron solution. The no-connections configuration (all the SMDs compute local) and the nearest-node configuration (each SMD connected to the closest RAT) are also taken into account for comparison. The throughput, the energy consumed by the SMDs and the time employed for accomplishing the application are evaluated and observed for each configuration, considering an increasing number of SMDs involved in the system, in particular 500, 1000, 2000, and 5000.

Table 5.1, Table 5.2, and Table 5.3 show, respectively, the results related to the throughput, the energy, and the time. The ex-post solution has been computed with Baron under the Neos server [1]. The QSAP model is implemented with the mathematical modeling language General Algebraic Modeling System (GAMS). The other methods have been implemented in Matlab.

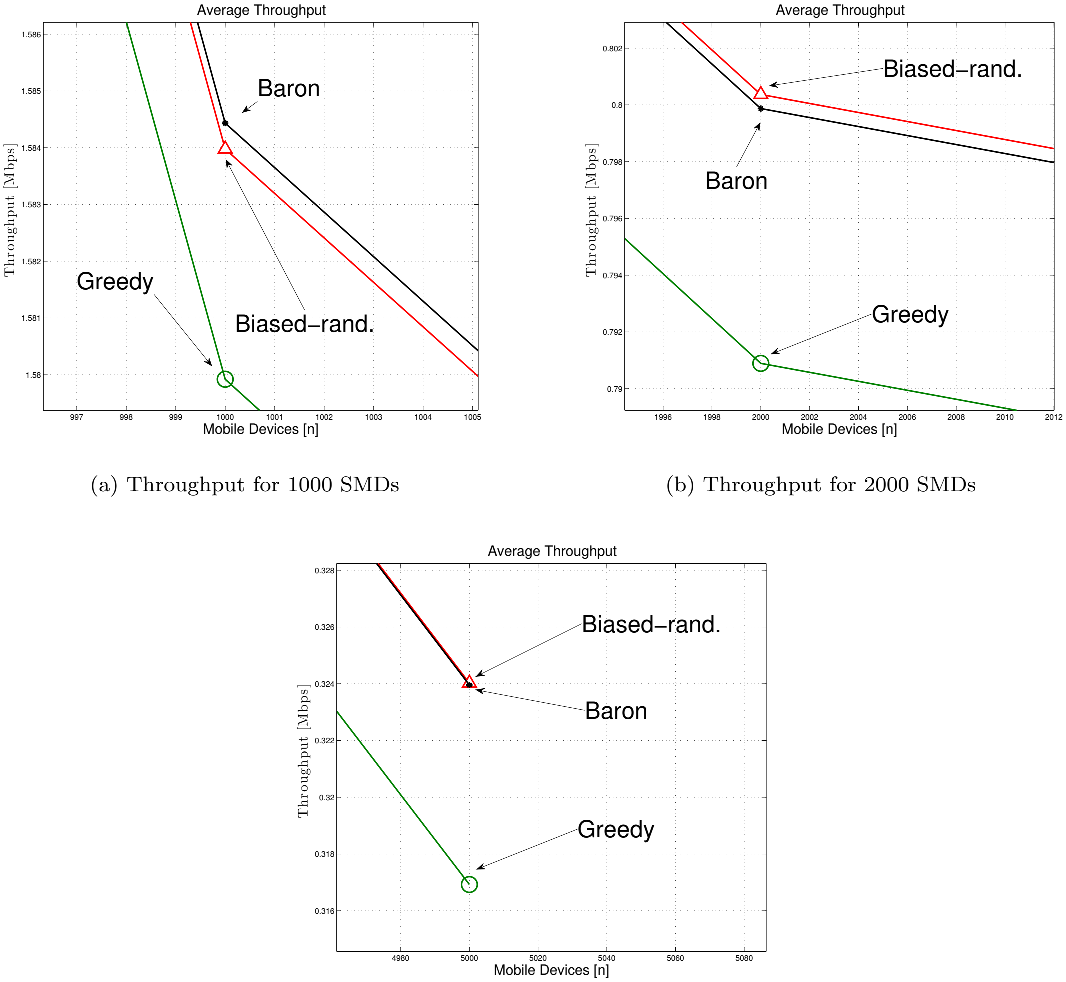
5.5 Analisys of Results

The results of Table 5.1, Table 5.2, and Table 5.3 show that the biased-randomized algorithm clearly improves the cell association configuration with respect to the greedy heuristic. These improvements are: 2.24% in throughput, 14.30% in energy consumption, and 0.42% in computation time.

Also, the average energy and time values provided by the randomized algorithm, which only requires some milliseconds to compute, are quite similar to the solutions provided by Baron, which

[1] http://www.neos-server.org

is stopped after 1,000 seconds of computation. The results are visually compared in Figure 5.3 and in 5.4a, where we can see that the trends of average throughput are very similar, practically overlapped, in the three cases of interest, i.e. the greedy heuristic, the biased randomized algorithm, and the Baron ex-post configuration. This means that the capacity of the RAT nodes is well-exploited in all the three cases considered. In Figure 5.3 the average throughput values are visualized in a different scale to make possible the comparison of the three algorithms.



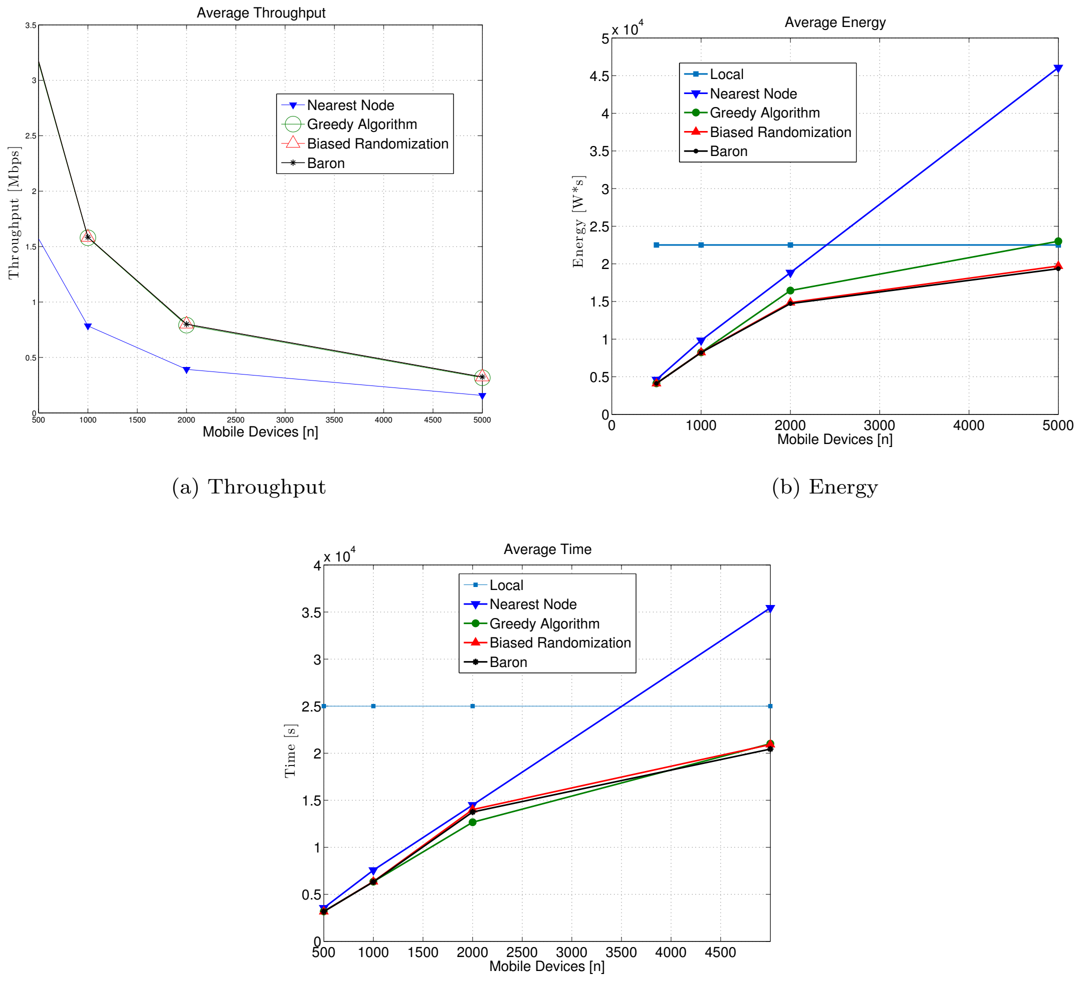
(a) Throughput for 1000 SMDs

(b) Throughput for 2000 SMDs

(c) Throughput for 5000 SMDs

Figure 5.3: Comparison of the average throughput

Considering the average energy consumed by a SMD to perform the application, which is the objective function to minimize, Figure 5.4b shows that the biased-randomized algorithm clearly outperforms the greedy heuristic. The same observation can be inferred from Figure 5.4c in relation to the average time that a device needs to accomplish the application. Both the average energy and the time values are very close to the respective Baron configuration average energy and time values, thus the biased randomized algorithm can be exploited by the SMDs to reach a near-optimal configuration in real time while the solution seen from a collective point of view minimizes the overall energy consumption. This is useful for offloading applications which need a fast cell association decision, for example when the users are in movement and the configuration must be updated very often.



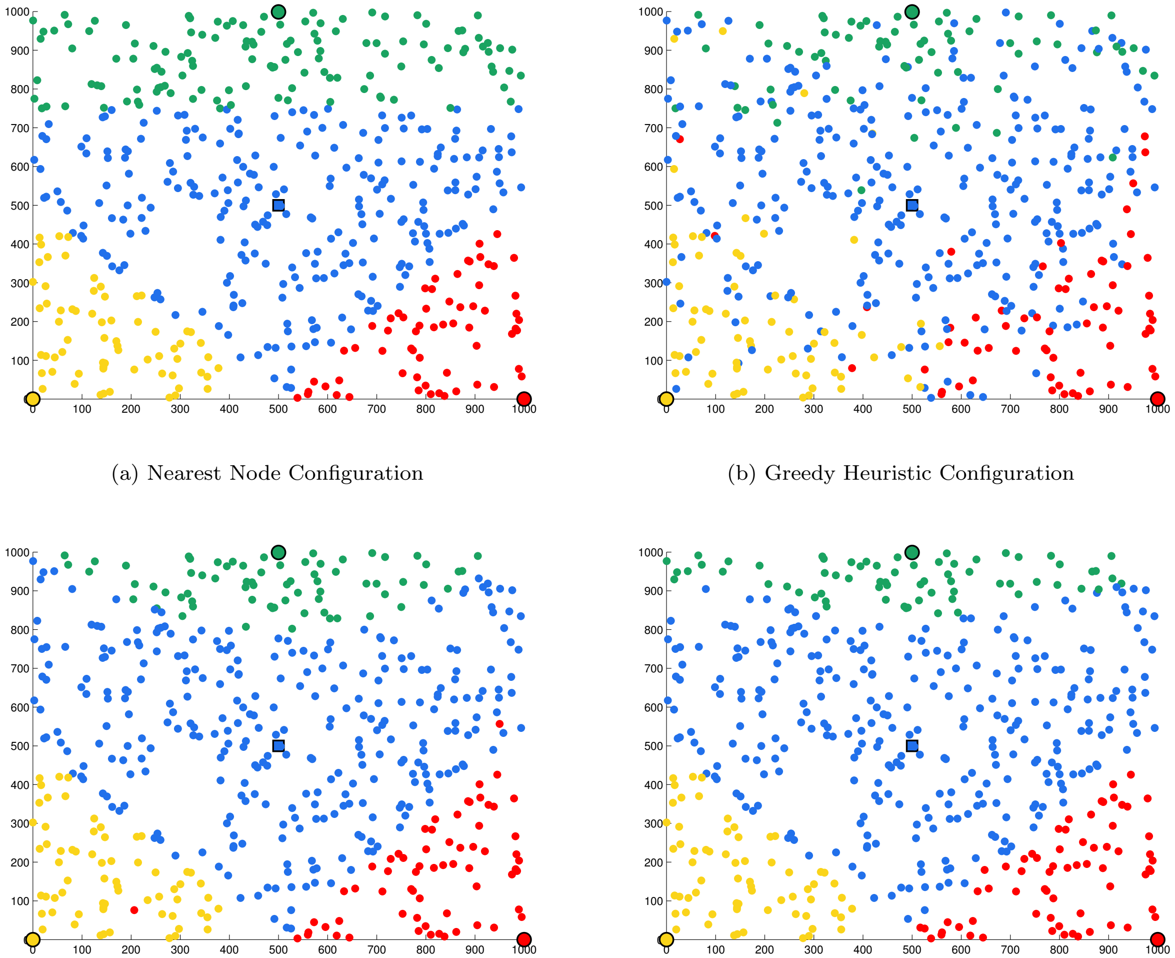
(a) Throughput

(b) Energy

(c) Latency

Figure 5.4: Average performance in terms of throughput, energy consumption and latency for a variable number of devices

Comparing the different configurations of the devices placed in the observed area, reported in Figure 5.5, Figure 5.6, Figure 5.7 and Figure 5.8 for a total number of SMD equal to 500, 1000, 2000 and 5000 respectively, we can note that the case referring to the biased-randomized algorithm is very similar to the ex-post optimal solution, where the SMDs connected to the same RAT are clustered in the neighborhood of the RAT.



(a) Nearest Node Configuration

(b) Greedy Heuristic Configuration

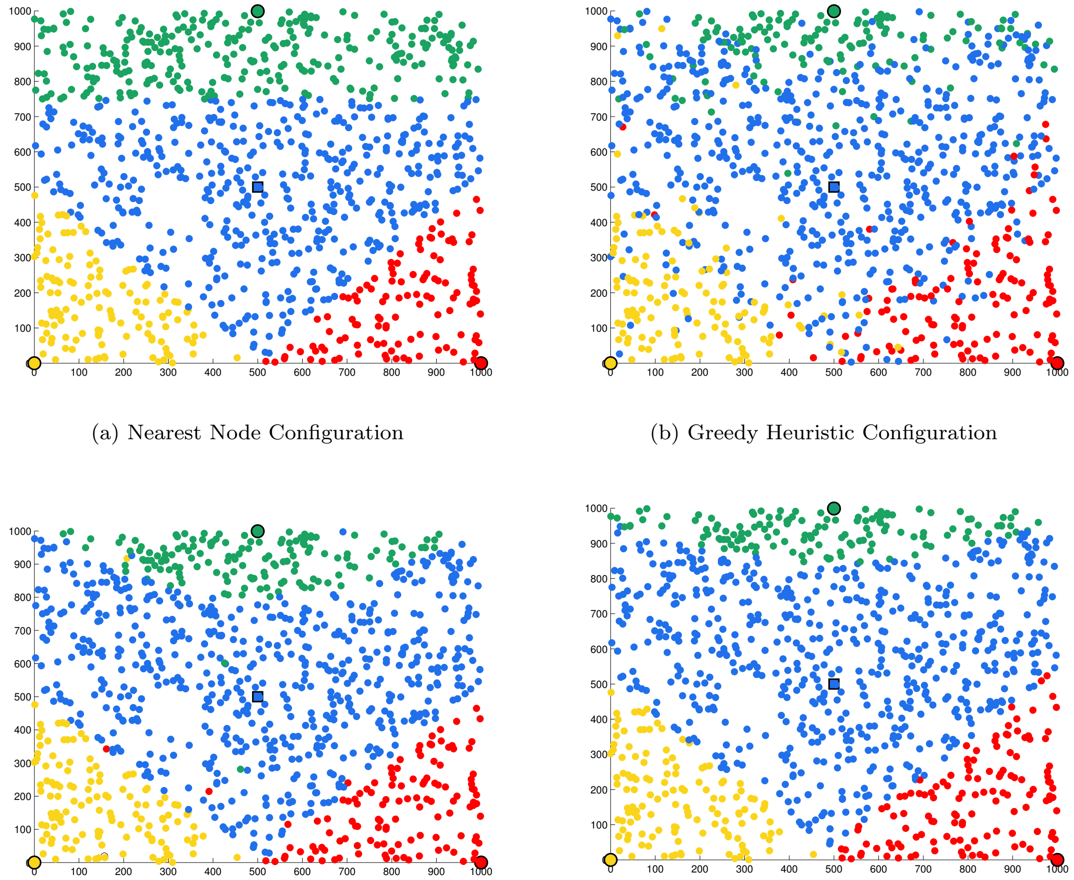
(c) Biased-randomized Configuration

(d) Ex-post Optimal Configuration

Figure 5.5: Cell Association in case of 500 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color.

5.6 Conclusions

In this chapter we presented different strategies for efficient cell association in pervasive wireless environments. In particular, we proposed an original probabilistic algorithm that allows to consider a



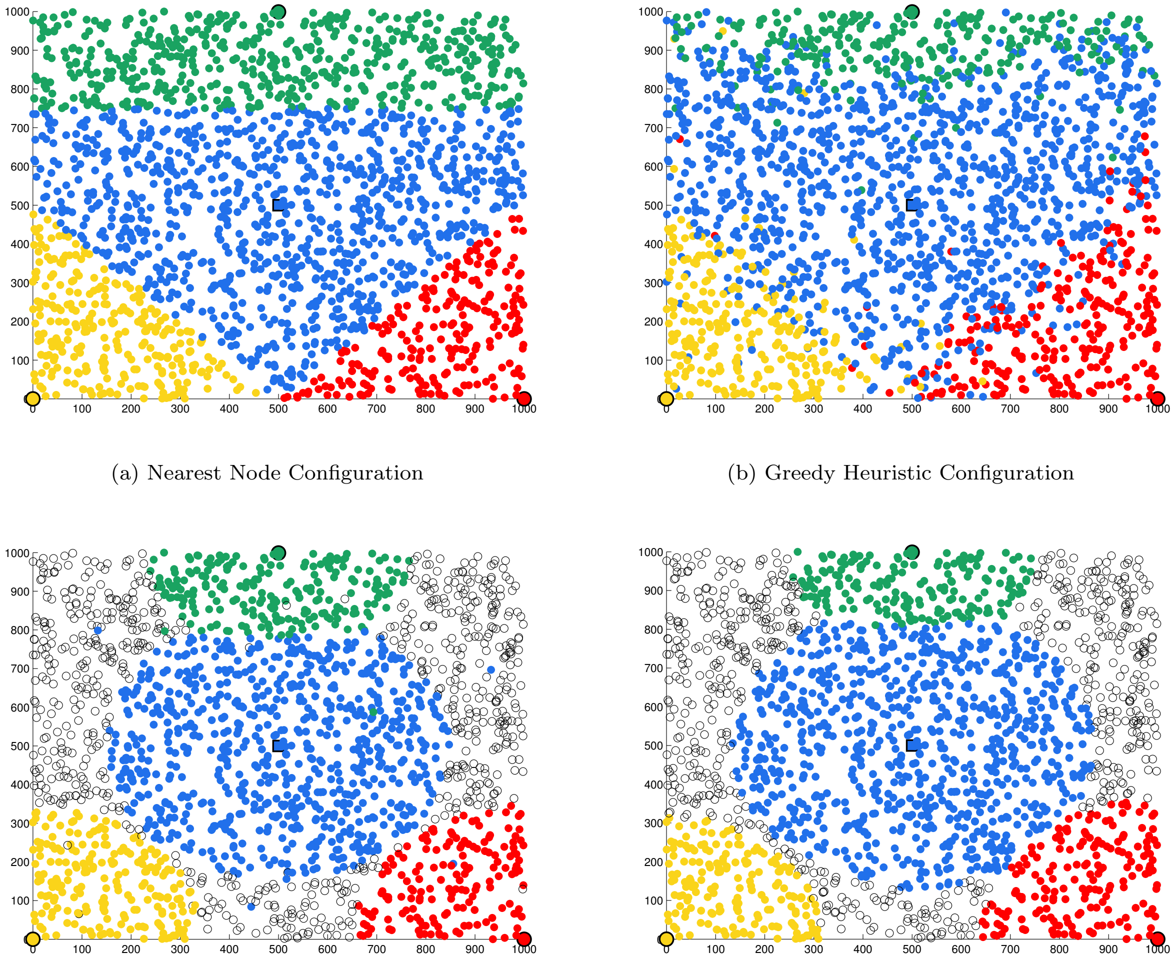
(a) Nearest Node Configuration

(b) Greedy Heuristic Configuration

(c) Biased-randomized Configuration

(d) Ex-post Optimal Configuration

Figure 5.6: Cell Association in case of 1000 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color.



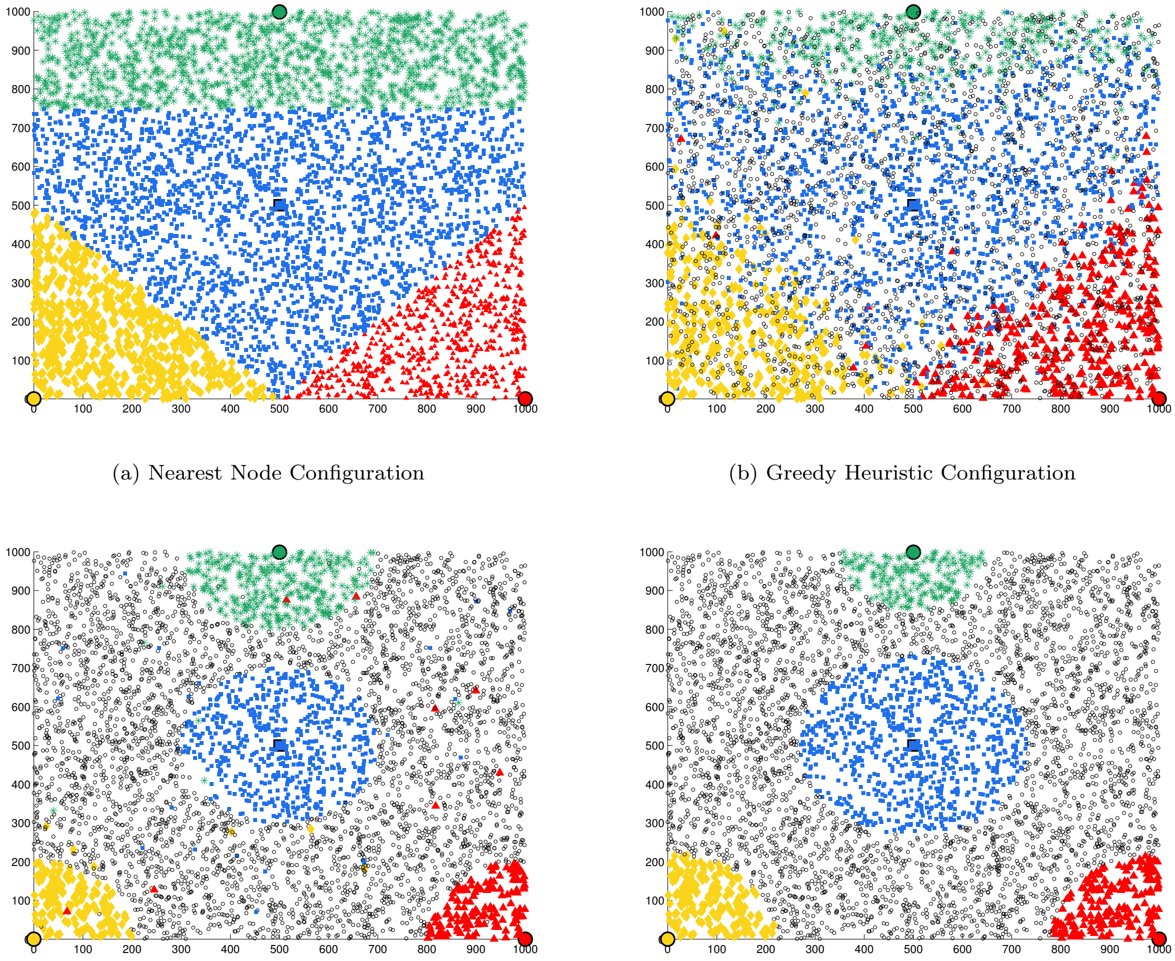
(a) Nearest Node Configuration

(b) Greedy Heuristic Configuration

(c) Biased-randomized Configuration

(d) Ex-post Optimal Configuration

Figure 5.7: Cell Association in case of 2000 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color.



(a) Nearest Node Configuration

(b) Greedy Heuristic Configuration

(c) Biased-randomized Configuration

(d) Ex-post Optimal Configuration

Figure 5.8: Cell Association in case of 5000 SMDs. Every color indicates that the SMD id connected to the RAT marked with the same color.

more global point of view –instead of just an individual one– during the real-time resource allocation. Thus, the mobile communication system is managed in a cooperative way, considering the QoS of the entire population of mobile users. This vision could be adopted for the provision of services in a smart city, improving performance and well-being of the community through a social use of the UMCC framework. Our approach is based on the use of biased-randomization techniques, which have been used in the past to solve similar combinatorial optimization problems in the fields of logistics, transportation, and production. This work extends their use to the field of smart cities and mobile telecommunications. Some numerical experiments contribute to illustrate the potential of the proposed approach.

Table 5.1: Throughput [Mbps]

N SMDs	500		1000		2000		5000	
	Avg.	Impr. [%]	Avg.	Impr. [%]	Avg.	Impr. [%]	Avg.	Impr. [%]
Nearest Node	1.5693	-50.26	0.7847	-50.33	0.3926	-50.36	0.1570	-50.46
Greedy	3.1551	ref.	1.5799	ref.	0.7909	ref.	0.3169	ref.
Biased Rand.	3.1699	+0.47	1.5840	+0.26	0.8004	+1.20	0.3240	+2.24
Baron	3.1707	+0.50	1.5844	+0.29	0.7999	+1.14	0.3240	+2.22

Table 5.2: Energy [W*s]

N SMDs	500		1000		2000		5000	
	Avg.	Impr. [%]	Avg.	Impr. [%]	Avg.	Impr. [%]	Avg.	Impr. [%]
Local	22500.0000	+445.26	22500.0000	+173.19	22500.0000	+36.80	22500.0000	-2.17
Nearest Node	4609.3141	+11.70	9819.6194	+19.23	18832.6714	+14.50	46060.3241	+100.26
Greedy	4126.5039	ref.	8236.0653	ref.	16447.8885	ref.	22999.8480	ref.
Biased Rand.	4106.4181	-0.49	8239.5161	+0.04	14848.2169	-9.73	19711.6979	-14.30
Baron	4104.4986	-0.53	8210.9327	-0.31	14716.6647	-10.53	19347.2418	-15.88

Table 5.3: Time for application computation [s]

N SMDs	500		1000		2000		5000	
	Avg.	Impr. [%]	Avg.	Impr. [%]	Avg.	Impr. [%]	Avg.	Impr. [%]
Local	25000.0000	+685.69	25000.0000	+294.13	25000.0000	+97.47	25000.0000	+19.01
Nearest Node	3553.3186	+11.67	7561.2457	+19.20	14494.3626	+14.49	35438.7108	+68.70
Greedy	3181.9261	ref.	6343.1271	ref.	12659.9142	ref.	21007.3416	ref.
Biased Rand.	3166.4755	-0.49	6345.7816	+0.04	14003.7438	+10.61	20918.6292	-0.42
Baron	3164.9989	-0.53	6323.7944	-0.30	13745.0152	+8.57	20432.3306	-2.74

CHAPTER 6

Conclusion and future works

6.1 Final Conclusions

The actual realization of the vision of smart cities is an exciting perspective with the potential of bringing tremendous improvements to our society, thanks to novel pervasive services assisting us in every aspect of our lives.

While the increasingly deep integration of ICT services in the physical world has been the main enabling factor motivating this vision, we believe that the intelligent and efficient exploitation of the infrastructures will be the key to its success.

Our contribution has proposed an original solution approach for dealing with common resources that need an optimization for the distribution among users.

We hope that our work was successful in proposing general and simple solution principles that can be used fruitfully as starting points for new research efforts toward the realization of future large scale smart pervasive environments. Finally, let us note that, while the scope of our research was focused on the specific application scenario of smart city environments, we believe that most of the presented results can be easily and successfully used in other scenarios with similar requirements of efficient and scalable distribution, such as the management of large scale telecommunication infrastructures or smart grids deployments.

6.2 Directions for Future Works

Cities around the globe are beginning to build out new digital services such as smart lighting, traffic, waste management and data analytics to reduce costs, tap new sources of revenue, create new innovation business districts and improve the overall quality of urban life. Integrated IoT and MCC applications enabling the creation of smart environments such as smart cities need to be able to combine services offered by multiple stakeholders and scale to support a large number of users in a reliable and decentralized manner. They deal with constraints such as access devices or data sources with limited power and unreliable connectivity. The UMCC needs to be enhanced to support a seamless execution of applications harnessing capabilities of multiple dynamic and heterogeneous resources to meet quality of service requirements of diverse users.

The management of distributes mobile platform resources need further investigation. This thesis is a starting point for a user-centric model that requires additional heterogeneity: i.e. customized needs and different QoS levels. For example, some users could not have problem of battery charge whereas others could need to save energy (i.e. SMD having a low-battery-level status). Thus, the clusterization debated in chapter 4 needs to be deeply investigated.

A further point of invetigation is the study of SMDs in motion, for analyzing the time evolution of the system: in chapter 5 the approach *on the fly* assigns a RAT node to a SMD every time a request occurs, but, since it depends on the position of the devices, the CA needs a time optimization subordinate to the speed of the devices.

In addition, considering an operational research perspective, the CA assignation in the UMCC can be seen as a stochastic allocation of resources, in particular an assignation of the SMDs to the available RATs, as shown in Figure 6.1.

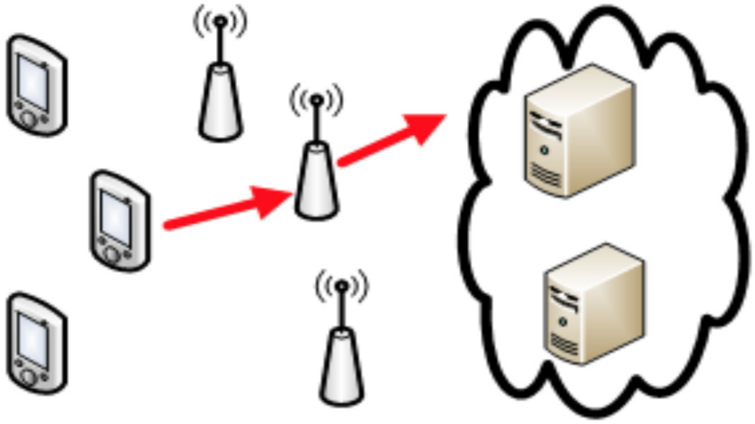


Figure 6.1: Allocation of mobile devices with random positions to RATs

In fact, being mobile, the position of the devices change over time, and it is possible to assume that the position of each device in a future target time can be characterized by two random variables, describing the x and y location, respectively as shown in Figure 6.2.

It will also be assumed that the expected value of each random variable will be given by the current position of the respective device. Assuming that the expected value of each random variable

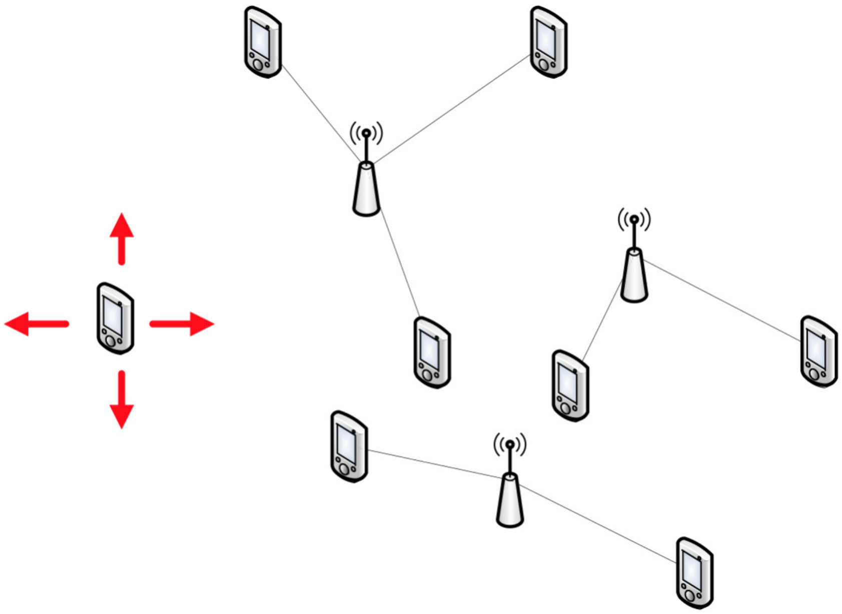


Figure 6.2: Random position of the devices in a target time

will be given by the current position of the respective device, it is possible to model this scenario as a Facility Location Problem (FLP) with stochastic assignment costs - i.e. a combination of different factors, such as time-based costs and energy-based costs, random variables proportional to the Euclidean distance between the device and the RAT node. It is worth to note that the RATs capacity could be considered either as virtually unlimited or not, thus leading to two different problems: the Uncapacitated FLP and the Capacitated FLP, respectively. One way to deal with these stochastic FLPs would be to use a simheuristic approach [59] i.e., a combination between a fast metaheuristic - able to quickly generate several good or promising solutions for the deterministic version of the FLP - and a fast simulation process able to estimate the expected cost associated to a given deterministic solution when it is used for the stochastic scenario. The main goal of this future work will be to minimize the expected costs of the assignment process. The simulation, however, could also provide insights on the robustness of each promising solution, enriching a risk analysis investigation.

Bibliography

[1] M. Dohler, C. Ratti, J. Paraszczak, and G. Falconer, "Smart cities [guest editorial]," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 70–71, Jun. 2013. vii

[2] B. Batty, *The New Science of Cities.* Cambridge, MA, USA: The MIT Press, 2013. vii

[3] G. E. Corazza, A. Vanelli-Coralli, and R. Pedone, "Technology as a need: Trends in the evolving information society," *Advances in Electronics and Telecommunications*, vol. 1, no. 1, pp. 124–132, 2010. 1

[4] S. Kachele, C. Spann, F. Hauck, and J. Domaschka, "Beyond IaaS and PaaS: An extended cloud taxonomy for computation, storage and networking," in *Proc. of 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC)*, Dresden, Germany, Dec. 2013, pp. 75–82. 1

[5] R. Fantacci, M. Vanneschi, C. Bertolli, G. Mencagli, and D. Tarchi, "Next generation grids and wireless communication networks: towards a novel integrated approach," *Wireless Communications and Mobile Computing*, vol. 9, no. 4, pp. 445–467, Apr. 2009. 1, 27, 39

[6] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 5, pp. 821–834, May 2012. 1

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, 2012. 1, 7, 49

[8] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Big Data and Internet of Things: A Roadmap for Smart Environments," vol. 546, pp. 169–186, 2014. 1, 7

[9] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013. 3, 27, 39

[10] W. Lumpkins, “The internet of things meets cloud computing [standards corner],” *IEEE Consum. Electron. Mag.*, vol. 2, no. 2, pp. 47–51, Apr. 2013. 4

[11] E. Lee, E.-K. Lee, M. Gerla, and S. Oh, “Vehicular cloud networking: architecture and design principles,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 148–155, Feb. 2014. 4

[12] M. Hasan, E. Hossain, and D. Niyato, “Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches,” *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013. 4

[13] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, “Communicating while computing: Distributed mobile cloud computing over 5g heterogeneous networks,” *Signal Processing Magazine, IEEE*, vol. 31, no. 6, pp. 45–55, Nov 2014. 5, 50

[14] N. Bhas, “Data offload ∼ connecting intelligently,” White paper, Juniper Research, Apr. 2013. 5

[15] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, “Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications,” *IEEE Commun. Mag.*, vol. 20, no. 3, pp. 14–22, Jun. 2013. 7

[16] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes, “Enabling Smart Cloud Services Through Remote Sensing: An Internet of Everything Enabler,” *Internet of Things Journal, IEEE*, vol. 1, no. 3, pp. 276–288, 2014. 7

[17] O. Vermesan and P. Friess, *Internet of Things - From Research and Innovation to Market Deployment.* Aalborg, DK: River Publishers, 2014. 16

[18] L. M. Correia and K. Wunstel, “Smart cities application and requirements,” White Paper, Net!Works European Technology Platform, May 2011. 16

[19] *Electromagnetic compatibility and Radio spectrum Matters (ERM); System Reference document (SRdoc): Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications*, ETSI Std. TR 103 055, 2011. 16

[20] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, “Toward cloud-based vehicular networks with efficient resource management,” *IEEE Netw.*, vol. 27, no. 5, pp. 48–55, Sep. 2013. 17, 56

[21] C. He, X. Fan, and Y. Li, “Toward ubiquitous healthcare services with a novel efficient cloud platform,” *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 230–234, Jan. 2013. 17

[22] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, “The role of cloudlets in hostile environments,” *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 40–49, Oct. 2013. 17

[23] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Commun. Mag.*, vol. 20, no. 3, pp. 34–44, Jun. 2013. 20, 25, 32, 39

[24] L. Badia, M. Lindström, J. Zander, and M. Zorzi, "An economic model for the radio resource management in multimedia wireless systems," *Comput. Commun.*, vol. 27, no. 11, pp. 1056–1064, Jul. 2004. 21, 42

[25] L. Jiao, R. Friedman, X. Fu, S. Secci, Z. Smoreda, and H. Tschofenig, "Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities," in *Proc. of FutureNetworkSummit 2013*, Lisboa, Portugal, Jul. 2013. 27, 39, 49

[26] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: Making smartphones last longer with code offload," in *Proc. of MobiSys '10*, San Francisco, CA, USA, Jun. 2010, pp. 49–62. 27

[27] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. of EuroSys '11*, Salzburg, Austria, Apr. 2011, pp. 301–314. 27

[28] X. Ma, Y. Zhao, L. Zhang, H. Wang, and L. Peng, "When mobile terminals meet the cloud: computation offloading as the bridge," *IEEE Netw.*, vol. 27, no. 5, pp. 28–33, Sep./Oct. 2013. 27

[29] M. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? The bandwidth and energy costs of mobile cloud computing," in *Proc. of IEEE INFOCOM 2013*, Turin, Italy, Apr. 2013, pp. 1285–1293. 27

[30] R. Murmuria, J. Medsger, A. Stavrou, and J. Voas, "Mobile application and device power usage measurements," in *Proc. of IEEE SERE 2012*, Jun. 2012, pp. 147–156. 27

[31] Y.-D. Lin, E.-H. Chu, Y.-C. Lai, and T.-J. Huang, "Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds," *IEEE Syst. J.*, 2013. 27

[32] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010. 27, 29, 30, 32, 41, 45, 75

[33] H. Wu, Q. Wang, and K. Wolter, "Tradeoff between performance improvement and energy saving in mobile cloud offloading systems," in *Proc. of IEEE ICC 2013 Workshops*, Budapest, Hungary, Jun. 2013, pp. 728–732. 27, 29, 30, 41

[34] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Computation offloading for mobile cloud computing based on wide cross-layer optimization," in *Proc. of FutureNetworkSummit 2013*, Lisboa, Portugal, Jul. 2013. 27

[35] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *IEEE Netw.*, vol. 27, no. 5, pp. 48–55, Sep. 2013. 32

[36] S. Pal, S. Das, and M. Chatterjee, "User-satisfaction based differentiated services for wireless data networks," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 2, May 2005, pp. 1174–1178 Vol. 2. 39, 42

[37] M. Amani, A. Aijaz, N. Uddin, and A. Aghvami, "On mobile data offloading policies in heterogeneous wireless networks," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, June 2013, pp. 1–5. 40

[38] D. Mazza, D. Tarchi, and G. E. Corazza, "A partial offloading technique for wireless mobile cloud computing in smart cities," in *Proc. of 2014 European Conference on Networks and Communications (EuCNC)*, Bologna, Italy, Jun. 2014. 40

[39] S. Lohier, A. Rachedi, and Y. Ghamri-Doudane, "A cost function for qos-aware routing in multi-tier wireless multimedia sensor networks," in *Proc. of the 12th IFIP/IEEE ICMMMNS*, 2009, pp. 81–93. 42, 43

[40] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012. 49

[41] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 974–983, April 2015. 50, 63

[42] A. Mtibaa, M. Abu Snober, A. Carelli, R. Beraldi, and H. Alnuweiri, "Collaborative mobile-to-mobile computation offloading," in *Proc. of 2014 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Miami,FL, USA, Oct. 2014, pp. 460–465. 50

[43] P. Si, F. Yu, and Y. Zhang, "Joint cloud and radio resource management for video transmissions in mobile cloud computing networks," in *Proc. of 2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 2270–2275. 50

[44] M. Anastasopoulos, A. Tzanakaki, G. Zervas, B. Rofoee, R. Nejabati, D. Simeonidou, G. Landi, N. Ciulli, J. Riera, and J. Garcia-Espin, "Convergence of heterogeneous network and IT infrastructures in support of fixed and mobile cloud services," in *Proc. of 2013 Future Network and Mobile Summit (FutureNetworkSummit)*, Lisbon, Portugal, Jul. 2013. 50

[45] E. A. Lee, J. Rabaey, B. Hartmann, J. Kubiatowicz, K. Pister, A. Sangiovanni-Vincentelli, S. A. Seshia, J. Wawrzynek, D. Wessel, T. S. Rosing, D. Blaauw, P. Dutta, K. Fu, C. Guestrin, B. Taskar, R. Jafari, D. Jones, V. Kumar, R. Mangharam, G. J. Pappas, R. M. Murray, and

A. Rowe, “The swarm at the edge of the cloud,” *IEEE Design Test*, vol. 31, no. 3, pp. 8–20, June 2014. 50

[46] J. Yang, S. Tilak, and T. S. Rosing, “A novel protocol for adaptive broadcasting of sensor data in urban scenarios,” *GLOBECOM - IEEE Global Telecommunications Conference*, pp. 1–6, 2013. 50

[47] P. Aghera, J. Yang, P. Zappi, D. Krishnaswamy, A. Coskun, T. S. Rosing, and T. S. Rosing, “Energy Management in Wireless Mobile Systems Using Dynamic Task Assignment.” 50

[48] D. Sabella, P. Rost, Y. Sheng, E. Pateromichelakis, U. Salim, P. Guitton-Ouhamou, M. di Girolamo, and G. Giuliani, “RAN as a service: Challenges of designing a flexible RAN architecture in a cloud-based heterogeneous mobile network,” in *Proc. of Future Network and Mobile Summit (FutureNetworkSummit), 2013*, Lisbon, Portugal, Jul. 2013. 55

[49] R. Stanica, E. Chaput, and A.-L. Beylot, “Local density estimation for contention window adaptation in vehicular networks,” in *Proc. of 2011 IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sept 2011, pp. 730–734. 56

[50] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, “Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture,” in *Proc. of 2012 IEEE Symposium on Computers and Communications (ISCC)*, Cappadocia, Turkey, Jul. 2012, pp. 59–66. 57

[51] A. A. Juan, J. Faulin, J. Jorba, D. Riera, D. Masip, and B. Barrios, “On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics.” 61, 64, 66, 72

[52] D. Mazza, D. Tarchi, and G. E. Corazza, “A user-satisfaction based offloading technique for smart city applications,” in *Proc. of IEEE Globecom 2014*, Austin, TX, USA, Dec 2014. 63, 66, 70, 75

[53] G. Clarke and J. W. Wright, “Scheduling of vehicles from a central depot to a number of delivery points,” *Operations Research*, no. 4, pp. 568–581, Jul. 64

[54] M. Nawaz, E. Enscore, and I. Ham, “A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem,” *Omega*, no. 1, pp. 91–95. 64

[55] “Computational experiments with algorithms for a class of routing problems,” *Computers & Operations Research*, vol. 10, no. 1, pp. 47 – 59, 1983. 64

[56] R. E. Burkard, “Quadratic assignment problems,” in *Handbook of Combinatorial Optimization*, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds. Springer New York, 2013, pp. 2741–2814. 70

[57] M. Tawarmalani and N. V. Sahinidis, “A polyhedral branch-and-cut approach to global optimization,” *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, 2005. 70

[58] T. A. Feo and M. G. C. Resende, “Greedy randomized adaptive search procedures,” *Journal of Global Optimization*, pp. 109–133. 72

[59] A. a. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira, “A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems,” *Operations Research Perspectives*, pp. 62–72. 89