UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Remote/Local offloading of mobile applications as an energy saving strategy

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Mauro Lucchini Depompa

In partial fulfilment

of the requirements for the degree in

Science and Telecommunication Technologies Engineering

Advisor: Olga Muñoz Medina and Antonio Pascual Iserte

Barcelona, January 2016

Abstract

The rapid progress of mobile computing becomes a powerful trend in development as well as in the commerce and industry areas. The limited resources affect significantly the improvement of some service quality. In this sense, offloading could result a way to improve the user experience since the user will be able to take advantage of more computing resources available in the network (for example in the access points) than just the ones that the own device offers. This project focuses on offloading techniques as an energy saving strategy that can allow increasing the battery lifetimes of mobile terminals.

The project concludes that the time needed for running certain processes locally at the mobile terminals is the most relevant factor to determine if offloading provides energetic benefits or not. The different variables involved that may affect the time needed to execute or either communicate are studied in this thesis so as to check the feasibility of the proposed techniques. Furthermore, the scope is opened to wider networks where computing entities can be located more remotely than at the access points.

Resum

El ràpid progrés de la computació mòbil s’ha convertit en una significativa tendència tan en l’àmbit del desenvolupament com en el del comerç i la indústria. La limitació de recursos afecta significativament a la millora de la qualitat dels serveis. En aquest sentit, l’*offloading* pot resultar una manera de millorar l’experiència de l’usuari, ja que li permet treure profit de recursos computacionals disponibles a la xarxa, per exemple en els punts d’accés, que no només els que li ofereix el seu propi dispositiu. Aquest projecte està focalitzat en la tècnica de l’*offloading* com a estratègia d’estalvi energètic, que permetrà incrementar la vida de les bateries dels terminals mòbils.

En el projecte es conclou que el temps necessari per executar localment cert procés és el factor més rellevant que determina si l’*offloading* proveeix beneficis energètics o no. Les diferents variables que poden afectar el temps necessari d’execució o bé de comunicació són estudiats en aquesta tesi amb la intenció de comprovar la viabilitat de la tècnica proposada. A més a més, s’estén l’estudi a xarxes més amplies on les entitats amb capacitats computacionals poden estar situades a punts més llunyans que en els propis punts d’accés.

Resumen

El rápido progreso de la computación móvil se ha convertido en una significativa tendencia tanto en el ámbito del desarrollo como en del comercio y la industria. Los recursos limitados afectan significativamente la mejora de la calidad de los servicios. En este sentido, el *offloading* puede resultar una manera de mejorar la experiencia del usuario dado que le permite sacar provecho de recursos de computación disponibles en la red, por ejemplo en los puntos de acceso, que no solo los que le ofrece su propio dispositivo. Este proyecto está focalizado en la técnica del *offloading* como una estrategia de ahorro energético que permitirá incrementar la vida de las baterías de los terminales móviles.

En el proyecto se concluye que el tiempo necesario para ejecutar localmente cierto proceso es el factor más relevante que determina si el *offloading* aporta beneficios energéticos o no. Las diferentes variables que pueden afectar el tiempo necesario de ejecución o comunicación son estudiadas en esta tesis con la intención de comprobar la viabilidad de la técnica propuesta. Además, se extiende el estudio a redes más amplias donde las entidades con capacidad computacional pueden estar situadas en puntos más lejanos que los propios puntos de acceso.

For all my dears who cheered me up and stand by me when I needed. This project is dedicated to all of you.

Acknowledgements

I need to leave a written record of how thankful I feel with the people who helped me to develop this project. I want to thank my advisors, Toni and Olga, who always have helped me with all the problems I came across along the development of the project. They were present in all the phases of the project and helped me not only with their useful advices and knowledge but also encouraging me.

It would be unfair not dedicating some words to my family and my couple, who were my upholder when everything seemed to be impossible. Sincerely, thanks for your support and patience.

Revision history and approval record

Revision	Date	Purpose
0	15/12/2015	Document creation
1	08/01/2016	Document revision
2	15/01/2016	Document revision
3	22/01/2016	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Mauro Lucchini Depompa	maurolucchinidepompa@gmail.com
Olga Muñoz Medina	olga.munoz@gmail.com
Antonio Pascual Iserte	antonio.pascual@gmail.com

Written by:		Reviewed and approved by:	
Date	15/01/2015	Date	22/01/2016
Name	Mauro Lucchini Depompa	Name	Olga Muñoz & Antonio Pascual
Position	Project Author	Position	Project Supervisors

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Statement of purpose

Mobile devices are increasingly playing an essential role in human life as one of the most effective and convenient communication tools not bounded by time and place. Mobile users accumulate rich experience from various services associated to mobile applications that can run on the devices and/or on remote servers via wireless networks. In this sense, there is a significant emerging convergence between cloud computing (CC) and wireless communication, providing consumers with access to a vast array of cloud applications and services with the convenience of anywhere, anytime and any network functionality from the device of their choice. Clearly, mobile cloud computing (MCC) is such a fruitful area for further research. Thanks to MCC, task offloading from smartphones to the cloud is a promising strategy to enhance the computing capability of smartphones and, in some cases, prolong their battery life due to the fact that fewer computations have to be carried out at the user terminals.

One key advantage of CC is that it enables resources and infrastructure to be shared among many users, and returned to a common resource pool when not needed. This offers flexibility in data provision, computation and storage, while allowing users to gain easy access to computing resources much more powerful than those provided by a single mobile terminal (MT).

CC has the potential to save mobile client energy but the savings from offloading the computation need to exceed the energy cost of the additional communication of the data through the wireless link. The trade-off point is strongly dependent on the energy efficiency of wireless communication and of local processing. Additionally, not only the amount of transferred data but also the traffic pattern is important, as well as the total delay associated to the transmission and/or processing of data. The purpose of this project is to mind this gap between the energy needed for communicating and for local executing at the MT. Beyond taking into account these factors, considering wider networks with computational entities located further than the access point (AP) location are crucial for the analysis too and this has also been considered in this project.

There are many factors to consider when thinking about MCC scenarios. The quantity of data defining the break-even for moving to cloud is highly dependent on the exact energy efficiencies of wireless communication and local computing. The target of this project is not finding a characteristic optimal value for any of the relevant factors in the system. The aim of the project is to open the scope as much as possible in order to obtain a general vision of the advantages and disadvantages of offloading as an energy saving strategy. Consequently, the analysis results obtained in the simulator are descriptors of several factors. This project observes the relevance of a wide range of variables present in the stage in terms of energy saving and technique feasibility, wishing obtaining attractive results to keep researching in this field in the future.

1.2. Requirements and specifications

Project requirements:

- A set of small cells (SCs) is able to communicate with the users.
- A set of VMs (VMs) is able to execute apps offloaded by the users.

- The SCs are connected with the VMs and the users with the SCs.
- The offloading manager is the responsible to take decisions concerning the offloading of the applications.
- The request for offloading can be rejected.

Project specifications:

- All the data will be processed in one place (either at the MT or in a VM). There is not data division.
- The user equipment will work with constant transmission power.
- Different APs use different frequencies. There is not interference between APs.
- The multiple access will be based on Time Division Multiple Access (TDMA) (for the transmission through the wireless channel, the transmission through the links between the APs and the VMs, and the allocation of the computational resources at the VMs or MTs).
- The latency and the bitrate of the backhaul network connecting APs and the VMs will be taken into account in the offloading decision. Whereas, latency is despised in radio communication between MTs and APs. In this document, it is considered that the latency is the time needed for the propagation of the signal itself through a link.
- All the users transmit with the same power.
- The user connects via wireless to just one single SC.

1.3. Context of the project

This project arises from the work done so far by some members of the SPCOM group, including my advisors Antonio Muñoz and Olga Muñoz, in the project TROPIC [1] ("Distributed computing, storage and radio resource allocation over cooperative femtocells"), funded by the European Commission. TROPIC proposes that smartphones offload the execution of some of their apps to the SCs, which are equipped with some computational resources. They are connected to them with the goal of decreasing the energy consumption and increasing the lifetime of the batteries of the terminals. In other words, this project aims at exploiting the convergence of pervasive femto-network infrastructure and CC paradigms for virtualisation and distribution of applications and services. The European Commission launched this project among others to explore the technological options available leading to the future generation of "wired" (optical) and "wireless" communications, with a view to having deployable systems by 2020.

SCs based networks are currently seen as the new communication paradigm for the ever-increasing ubiquitous wireless traffic demands. Being pervasive by nature, its proximity to the subscriber opens a new world of possibilities for the development of applications. Among them, CC services demanded by smartphones could be moved from large server farms to SCs, provided that these are equipped with computational and storage resources, thus improving user experience on latency and download/upload speed.

The current project is an extension of the exposed above in the sense that TROPIC has been the motivation to keep on working on this field. Therefore, some of the background has been TROPIC, which, in addition, has inspired our work. Now, the purpose is to extend the research on offloading techniques already developed that have been proved to be feasible in some of the networks and communications resources available nowadays. By far, this project is not aiming at researching a completely new topic that has never been on the scope. It is designed to work on issues that have been largely discussed before so as to contribute to the implementation and deployment of these specific techniques.

While TROPIC considers all the VMs executing offloaded apps are in the APs, in this project more general scenarios will be considered. Particularly, VMs can be located in the APs or in far clouds. Due to this fact, the optimization of offloading, in terms of choosing the VMs, will take into account simultaneously: the VMs computational power, associated latencies to use certain VMs, radio channels availability and energetic aspects.

It is important to remark that efficient wireless resource management methods are required to provide Quality-of-Service (QoS) guarantee for the transmission of data in order to make use of cloud services. In this situation, once the environment is described, an optimal solution will be given to provide this service to those terminals who are desiring to offload their running applications so as to increase their battery life-time.

1.4. Work plan

1.4.1. Work packages

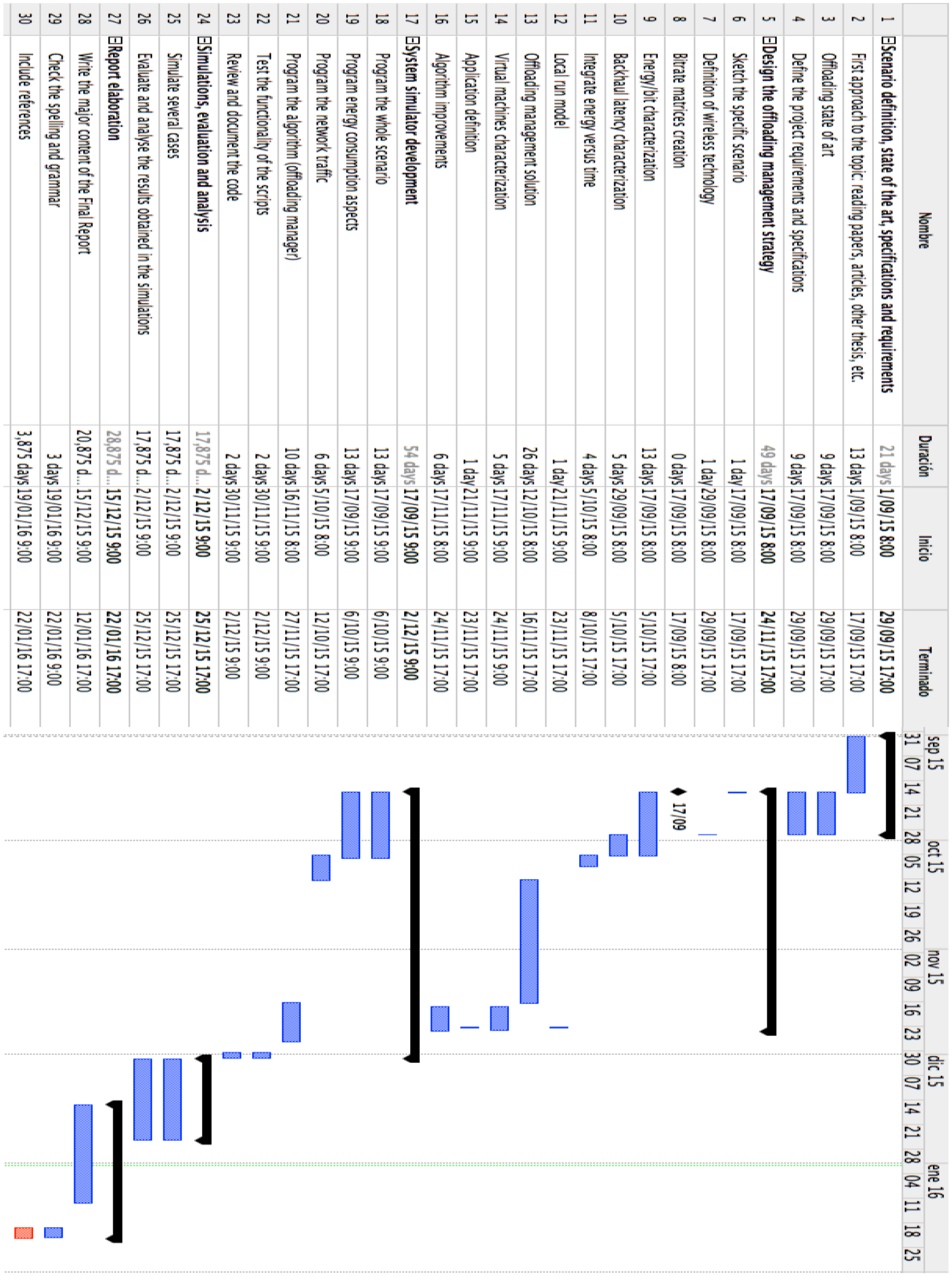
Project: Scenario definition, state of the art, specifications and requirements	**WP ref: WP1**	
Major constituent: Research		
Short description: Check the state of the art that will become the basis to set up the environment.	Planned start date: 01/09/15 Planned end date: 29/09/15	
	Start event: 01/09/15 End event: 29/09/15	
Internal task T1: First approach to the topic: reading papers, articles, other thesis, etc. Internal task T2: State of art in relation with offloading strategies. Internal task T3: Definition of the project requirements and specifications.	Deliverables: Project Proposal and Work plan	Dates: 05/10/15
Project: Design of the offloading management strategy	**WP ref: WP2**	
Major constituent: Algorithm research		
Short description: Work on providing an algorithm to manage the traffic.	Planned start date: 17/09/15 Planned end date: 24/11/15	

The offload manager will use this algorithm to control the network resources available.	Start event: 17/09/15 End event: 24/11/15	
Internal task T1: Sketch of the specific scenario considering the requirements & specifications agreed. Internal task T2: Definition of wireless technology used and its specifications (bandwidth characterization). Internal task T3: Bitrate matrices creation for APs and backhaul connections. Internal task T4: Energy/bit characterization for the connections available. Internal task T5: Backhaul latency characterization. Internal task T6: Integration energy versus time consuming offloading solutions. Internal task T7: Modelling of the option of running the app in local in terms of energy and time consuming. Internal task T8: Work out a solution for offloading management. Internal task T9: Characterization of the VMs. Internal task T10: Definition of the application. Internal task T11: Review and research for possible better options.		
Project: System simulator development	**WP ref: WP3**	
Major constituent: Programming		
Short description: Program the algorithm developed in the previous work package.	Planned start date: 17/09/15 Planned end date: 02/12/15	
	Start event: 17/09/15 End event: 04/12/15	
Internal task T1: Programming of the whole scenario (devices and connections). Internal task T2: Programming of the aspects related to device energy consumption. Internal task T3: Programming of the network traffic to insert into the previous script (events generation injection). Internal task T4: Program the algorithm (offloading manager). Internal task T5: Test of the functionality of the scripts. Internal task T6: Review and document the code.	Deliverables: Critical Review	Dates: 01/12/15

Project: Simulations, evaluation and analysis	**WP ref: WP4**	
Major constituent: Simulation		
Short description: Complete simulations will be run to evaluate the performance of the proposed offloading procedure in different scenarios.	Planned start date: 02/12/15 Planned end date: 22/12/15	
	Start event: 02/12/15 End event: 25/12/15	
Internal task T1: Simulation of several cases. Internal task T2: Evaluation and analysis of the results obtained in the simulations.		
Project: Report elaboration	**WP ref: WP5**	
Major constituent: Documentation		
Short description: The project work will be reported so as to generate the final document.	Planned start date: 02/12/15 Planned end date: 22/01/16	
	Start event: 15/12/15 End event: 12/01/16	
Internal task T1: Writing of the major content of the Final Report. Internal task T2: Checking of the spelling and grammar. Internal task T3: Including references.	Deliverables: Final Report	Dates: 25/01/16

Table 1: Work packages and tasks

1.4.2. Gantt diagram



	Nombre	Duración	Inicio	Terminado
1	⊟Scenario definition, state of the art, specifications and requirements	21 days	1/09/15 8:00	29/09/15 17:00
2	First approach to the topic: reading papers, articles, other thesis, etc.	13 days	1/09/15 8:00	17/09/15 17:00
3	Offloading state of art	9 days	17/09/15 8:00	29/09/15 17:00
4	Define the project requirements and specifications	9 days	17/09/15 8:00	29/09/15 17:00
5	⊟Design the offloading management strategy	49 days	17/09/15 8:00	24/11/15 17:00
6	Sketch the specific scenario	1 day	17/09/15 8:00	17/09/15 17:00
7	Definition of wireless technology	1 day	29/09/15 8:00	29/09/15 17:00
8	Bitrate matrices creation	0 days	17/09/15 8:00	17/09/15 8:00
9	Energy/bit characterization	13 days	17/09/15 8:00	5/10/15 17:00
10	Backhaul latency characterization	5 days	29/09/15 8:00	5/10/15 17:00
11	Integrate energy versus time	4 days	5/10/15 8:00	8/10/15 17:00
12	Local run model	1 day	21/11/15 9:00	23/11/15 17:00
13	Offloading management solution	26 days	12/10/15 8:00	16/11/15 17:00
14	Virtual machines characterization	5 days	17/11/15 9:00	24/11/15 9:00
15	Application definition	1 day	21/11/15 9:00	23/11/15 17:00
16	Algorithm improvements	6 days	17/11/15 8:00	24/11/15 17:00
17	⊟System simulator development	54 days	17/09/15 9:00	2/12/15 9:00
18	Program the whole scenario	13 days	17/09/15 9:00	6/10/15 9:00
19	Program energy consumption aspects	13 days	17/09/15 9:00	6/10/15 9:00
20	Program the network traffic	6 days	5/10/15 8:00	12/10/15 17:00
21	Program the algorithm (offloading manager)	10 days	16/11/15 8:00	27/11/15 17:00
22	Test the functionality of the scripts	2 days	30/11/15 9:00	2/12/15 9:00
23	Review and document the code	2 days	30/11/15 9:00	2/12/15 9:00
24	⊟Simulations, evaluation and analysis	17,875 d...	2/12/15 9:00	25/12/15 17:00
25	Simulate several cases	17,875 d...	2/12/15 9:00	25/12/15 17:00
26	Evaluate and analyse the results obtained in the simulations	17,875 d...	2/12/15 9:00	25/12/15 17:00
27	⊟Report elaboration	28,875 d...	15/12/15 9:00	22/01/16 17:00
28	Write the major content of the Final Report	20,875 d...	15/12/15 9:00	12/01/16 17:00
29	Check the spelling and grammar	3 days	19/01/16 9:00	22/01/16 9:00
30	Include references	3,875 days	19/01/16 9:00	22/01/16 17:00

Figure 1: Gantt diagram

1.5. Deviations from the initial plan

Initially, the project was deliberately focused on offloading of mobile applications to clusters of SCs. But while the software was being built, it enabled computational entities to be also located out of the APs. This fact concerned us that there were many topics that were taking centre stage and the study was not located just in SCs. What is more, the motivation of reducing energy consumption has been growing since the beginning of the project even to become the main protagonist of the project.

2. State of the art: Mobile cloud computing

CC has been widely recognized as the next generation computing infrastructure. It offers some advantages by allowing users to use infrastructure (e.g., servers, networks, and storages), platforms (e.g., middleware services and operating systems), and software (e.g., application programs) provided by cloud providers at low cost. Mobile applications can be rapidly provisioned and released with the minimal management efforts or service provider's interactions. With the explosion of mobile applications and the support of CC for a variety of services for mobile users, MCC is introduced as an integration of CC into the mobile environment. MCC brings new types of services and facilities to mobile users to take full advantages of CC.

2.1. Overview of mobile cloud computing

MCC at its simplest refers to an infrastructure where both the data storage and data processing happen outside of the MT. MCC moves the computing power and data storage away from mobile phones into the cloud, bringing applications and mobile computing to not just smartphone users but also a much broader range of mobile subscribers.

2.2. Mobile cloud computing architectures

According to [2], in a general architecture of MCC, MTs are connected to the mobile networks via base stations that establish and control the connections (air links) and functional interfaces between the network and the MTs. Mobile users' requests and information are transmitted to the central processors that are connected to servers providing mobile network services. Here, mobile network operators can provide services to mobile users as authentication, authorization, and accounting based on the home agent and subscribers' data stored in databases. After that, the subscribers' requests can be delivered to a cloud through the Internet. In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services. These services are developed with the concepts of utility computing, virtualization, and service-oriented architecture (e.g., web, application, and database servers). In addition, these services can be provided in the APs in case they have computational and storage abilities.

2.3. Advantages of mobile cloud computing

This section points out the principal advantages of MCC:

- **Extending battery lifetime:** Battery is one of the main concerns for mobile devices. Several solutions have been proposed to enhance the CPU performance and to manage the disk and screen in an intelligent manner to reduce power consumption. However, these solutions require changes in the structure of mobile devices, or they require a new hardware that results in an increase of cost and may not be feasible for all mobile devices. Computation offloading technique is proposed with the objective to migrate the large computations and complex processing from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This avoids taking a long application execution time on mobile devices that results in large amount of power consumption.

- **Improving data storage capacity:** Storage capacity is also a constraint for mobile devices. MCC is developed to enable mobile users to store/access large quantities of data in the cloud through wireless networks. CC can efficiently support various tasks for data warehousing, managing and synchronizing multiple documents online.

- **Improving processing power:** MCC also helps in reducing the running cost for compute-intensive applications that take long time. Powerful machines can solve complex problems much faster than MTs could ever do because of their computational power differences. MCC can provide a better user experience reducing time of execution, especially in mobile edge computing. For instance, the computational entity is located in the AP and it can execute the offloaded process.

- **Improving reliability:** Storing data or running applications in clouds is an effective way to improve the reliability because the data and application are stored and backed up on a number of computers. This reduces the chance of data and application lost in the mobile devices. In addition, MCC can be designed as a comprehensive data security model for both service providers and users.

2.4. Approaches of mobile cloud computing

In this section two different lists are exposed in order to leave a written record of relevant issues related to MCC in terms of mobile communications and computing side.

2.4.1. Mobile communications

- **Low bandwidth:** Bandwidth is one of the big issues in MCC because the radio resource for wireless networks is much more scarce as compared with the traditional wired networks.

- **Availability:** Service availability becomes a more important issue in MCC than that in the CC with wired networks. Mobile users may not be able to connect to the cloud to obtain a service due to traffic congestion, network failures, and being out of the coverage area.

- **Heterogeneity:** MCC will be used in highly heterogeneous networks in terms of types of wireless network interfaces. Different mobile nodes access the cloud through different radio access technologies such as GPRS, WiMAX and WLAN. As a result, an issue related to how to handle the wireless connectivity while satisfying MCC requirements arises (e.g., always-on connectivity, on-demand scalability of wireless connectivity, and the energy efficiency of mobile devices).

2.4.2. Computing side

- **Computing offloading in static and dynamic environments**

  Offloading in the static environment (without changing connection status and bandwidth) is not a common scenario, it will not be static most of the time. For a code compilation, offloading might consume more energy than that of local processing when the size of codes is small. Therefore, it is a critical problem for mobile devices to determine whether to offload and which portions of the codes of the applications need to be offloaded to improve the energy efficiency.

  When offloading in a dynamic network environment (changing connection status and bandwidth), the environment changes can cause additional problems. For example, the transmitted data may not reach the destination, or the data executed on the server will be lost when it has to be returned to the sender. What is more, offloading decision depends mainly on the bandwidth and the connection status. Therefore, the decision-making is much more challenging in dynamic environments. Even to the extent that a decision could become inopportune if the environment changes sharply.

- **Security**

  Protecting user privacy and data/application secrecy from adversary is a key to establish and maintain consumers' trust in the mobile platform, especially in MCC. This project does not discuss security and privacy issues.

3. Methodology / project development

3.1. Introduction to the simulator

This project consisted on the development of a simulator that describes a whole environment where users can offload their computation requests. The principal aim was to build a tool to run simulations that help to obtain conclusions about the usage of the offloading technique.

This simulator is available in my personal git-hosting repository [3]. The code is shared there to let readers understand deeply how the scenario is created and how the tests are carried out. The comments in the code describe carefully the behaviour of each component in the software.

3.2. Scenario modelling

The first step to model the scenario is to draw the sketch exposed in Figure 2. There are three stages: MTs, APs (some of them with attached VMs) and remote VMs. Users will hold MTs that will try to offload when offloading provides energy saving. Via wireless, they will connect to a visible AP. Routing the request to a VM is the only task assigned to APs. Then, VMs are the components that have computational power to process requests. They can be located next to APs (local VMs) or in further locations through the backhaul (remote VMs). In the case that the VM is located in the AP, there is no backhaul because the AP and the computational entity is physically located in the same place so they are directly connected. Otherwise, the backhaul has such an important role in the scenario configuration.

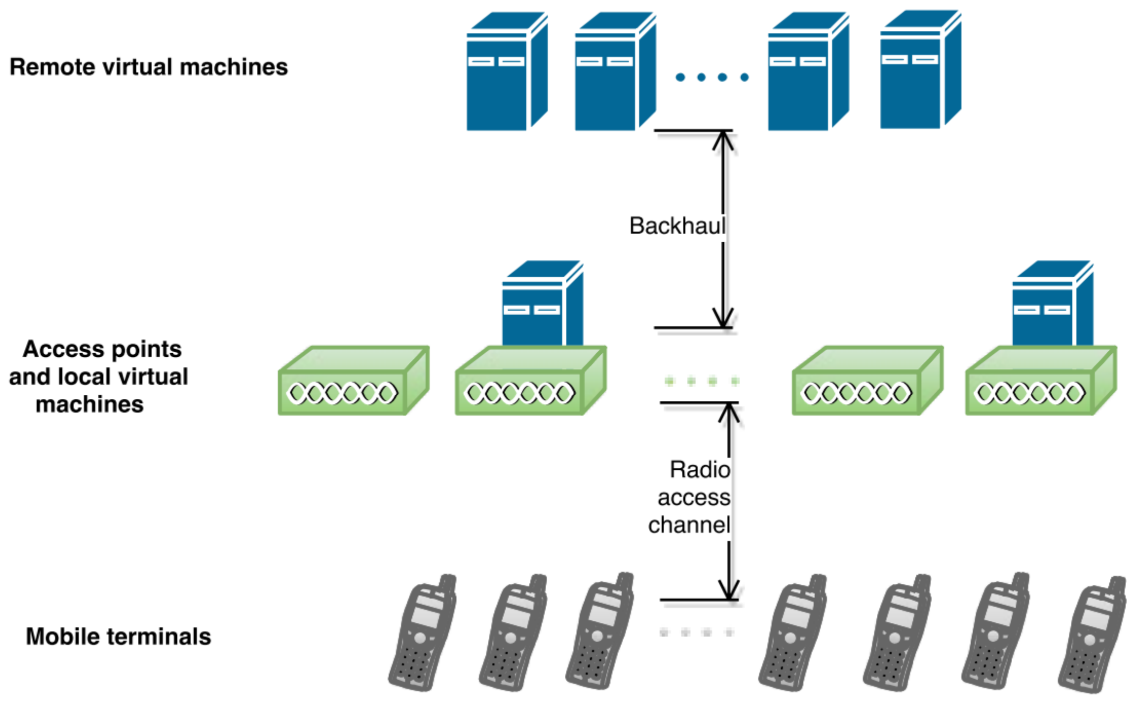


Figure 2: Scenario design

3.2.1. Radio access channel

To model the radio access channel, the system gives a random communication rate between the MTs and the APs so as to model uplink (UL) and downlink (DL). The communication rates between the MTs and the APs depend on the channel conditions between transmitters and receivers. Therefore, not all the MTs can connect to all the APs in the scenario and vice versa. This is simulated as a zero communication rate.

The radio access channel has been characterized as if the technology in presence for all the users was LTE. Consequently, the assigned rates meet the standard of this technology. To do so, first an index is randomly assigned to each user that will be used to point to one of the Modulation & Coding Scheme (MCS) [4] in order to perform the calculus of the radio rate (1).

Index	MCS	Index	MCS	Index	MCS
1	0.1523	6	1.1758	11	3.3223
2	0.2344	7	1.4766	12	3.9023
3	0.3770	8	1.9141	13	4.5234
4	0.6016	9	2.4063	14	5.1152
5	0.8770	10	2.7305	15	5.5547

Table 2: Modulation Coding Scheme indexing

$$rate = MCS * \frac{84}{0.5*10^{-3}} * 100 * \varepsilon \qquad (1)$$

In formula (1), MCS is the value assigned from Table 2. There are 84 resource elements in one physical resource block (PRB) of half millisecond. Working at the 20MHz band, like in our model, there are 100 PRB. Finally, all this calculus has to be weighted with ε that is the percentage of useful data in transmission. In this case, $\varepsilon = 0.8895$.

The calculus previously exposed allows building two connectivity matrices: one for the UL and one for the DL. Each component of these matrices contains a communication rate characterizing each pair of MT-AP. For example, the following UL connectivity matrix M contains the communication rates of K MTs and N APs. MT "x" and AP "y" are connected with a rate $M(x, y)$.

$$M = \begin{pmatrix} r_{1,1} & \cdots & r_{1,N} \\ \vdots & \ddots & \vdots \\ r_{K,1} & \cdots & r_{K,N} \end{pmatrix}$$

The UL and the DL are characterized in the same way. This approximation is enough to characterize the radio channel in our model.

The transmission through the radio channel is based on TDMA.

3.2.2. Backhaul network

In order to model the backhaul, two new connectivity matrices are created. These matrices are composed like the ones created for radio access channel but now they contain the rates to each combination between all APs and all VMs. This does not mean that this is a peer-to-peer system. Some of the connections will be assigned a zero transmission rate. In other words, the communication between these APs and VMs does not exist. Though, some of them will be assigned with a theoretical infinite transmission rate indicating that there is no backhaul connecting the VM and the AP because they are physically located in the same place. Obviously, for these cases, the latency (2) will be zero.

$$latency = \frac{d*n}{c} \qquad (2)$$

where d is the length of the backhaul link, c is the light velocity and n is the refractive index.

In this occasion, the technology in use for each connection is assigned with indexes for each pair AP-VM (see Table 3). After, a realistic rate to this certain technology is associated and replaced in the connectivity matrices. The structure of forward and backwards matrices is the same that for the UL and DL in the radio access channel but with N APs and M VMs. The model considers symmetrical forward and backwards directions in terms of rates, so in reality there could be just one matrix that fixes the communication rate for both ways. However, two matrices will be created so as to be able to model asymmetrical configuration in future research.

Index	Backhaul Technology	Throughput
1	Fibre Access 1	10M-10 Gbps
2	Fibre Access 2	100-1000 Mbps
3	Fibre Access 3	50M-10 Gbps
4	DSL Access	10-100 Mbps
5	Cable	10-100 Mbps
6	Wireless Backhaul	10 Mbps – 100 Mbps

Table 3: Backhaul technologies

In parallel, since the VMs may be located physically in remote locations, we have to model also the component of delay related to the latency (i.e., the time needed for the propagation of the signal itself through the backhaul link). In order to take this account, we create another matrix that will contain the latencies associated to each connection. VMs can be allocated to any of the locations exposed in Table 4. Once the places are known, the latency is calculated and assigned to each connection (2). Factor $n \geq 1$ since optical fibre could be one of the technologies in use [5].

City	Distance (km)	Latency (s)
Barcelona	30	0.031
Madrid	620	0.649
Paris	1,035	1.083
London	1,486	1.555
Amsterdam	1,533	1.605
Vienna	1,828	1.913
Warsaw	2,397	2.509
Washington	6,489	6.792
Miami	7,542	7.894
Dallas	8,330	8.719
Los Angeles	9,653	10.103
Hong Kong	10,053	10.522
Tokyo	10,413	10.899
Singapore	10,875	11.383
Sydney	17,179	17.981

Table 4: Distance and latency of city locations

The transmission in the backhaul is also based on TDMA.

3.2.3. Mobile terminal characterization

When it comes to characterize the terminal in terms of computational power, several factors can be taken into account. However, for our model is enough to have different options available in order to compare and contrast what happens when devices can resolve requests at slow or high velocity. This is the reason why the simulator does not deep into a sophisticated model to characterize the functionalities of a mobile and the factors that contribute the speed for each kind of application.

In the simulator we have assumed that there are five kinds of mobile phones. They basically differ on how fast they process the bits (bits/s), see Table 5.

	Rate for application of kind A (Mb/s)	Rate for application of kind B (Mb/s)
MT 1	2.6	4
MT 2	1.05	1.62
MT 3	0.79	1.22
MT 4	0.62	0.96
MT 5	0.3	0.47

Table 5: MT processing rates

The values of Table 5 are obviously an approximation of the real values. These computation rates characterize MT computational power. The importance of providing these values is to be able to characterize the time the device is processing and not for the speed itself. In fact, just the first device has been characterized with real values of an existing mobile phone [6]. The rest of device values were obtained extrapolating by a common benchmark [7].

There are two columns with different rates to emphasize that for different kinds of applications, depending what they do, these exemplary rates may vary. In this case in Table 5, MT 1 is the most powerful device whereas MT 5 is the weakest. It is accomplished for the heavy process (Application of kind A) and for the light process (Application of kind B).

MT occupation by applications is also based on TDMA. This means that apps cannot be run in parallel or in other words the computational resources are allocated to apps in a time division fashion.

3.2.4. Remote machines characterization

The principal aim of VM characterization is to model how much time the machine needs to process the load. Depending on the application complexity, for the same machine, it would take more or less time. Again, the model to characterize a VM and the factors that contribute to the speed can be as simple as for the MT.

The most important value to set is the factor that relates how faster these machines are with respect to MTs. To get a fair comparison, using a benchmark is required. The benchmark used for comparing the MTs and the processors available in the VMs is exposed in Table 6 according to Passmark [8]. The values in Table 6 are all synthetic, i.e., their meaning just has sense if the performance is compared, so these values do not have units. Then, with the help of this benchmark, Table 7 has been used to characterize the process executions in VMs.

	Passmark score
MT 1	7,494
MT 2	3,031
MT 3	2,281
MT 4	1,791
MT 5	877
VM 1	19,491
VM 2	8,300
VM 3	7,983

Table 6: Passmark scores

	Rate for application of kind A (Mb/s)	Rate for application of kind B (Mb/s)
VM 1	6.76	10.4
VM 2	2.88	4.43
VM 3	2.77	4.26

Table 7: VM processing rates

VM occupation by applications is also based on TDMA. This means that apps cannot be run in parallel or in other words the computational resources are allocated to apps in a time division fashion.

3.2.5. Load monitoring

The activity of the users with MTs is modelled as an arrival Poisson process.

There are different inputs to set:

- Number of users
- Simulation time
- Size of packages
- Offered load: Average number of requests per second generated by each user. The average number of requests per second per user is considered the same for all the users.

The output of the event generator is a matrix of three rows and N columns, where N is the number of requests generated. In the first row, we include the time stamp when the request was created. In the second raw, there is the index indicating which user generated this request. Finally, in the third raw, the amount of bits associated to the request is annotated. This number of bits is set with the input value used for calculating a random number from an exponential distribution. Nevertheless, this has been modified depending on the test run. In some cases the packages are the same value for all requests.

3.2.6. Energy models

The energy models refer to the consumption in the mobile phones that will decide to offload or not. The energy consumption at the network side, that is, at the APs or VMs, will not be taken into account in this project. What matters for us is only the energy of the MT that the user is holding.

Reasonably, energy models play an important role in this study since the algorithm to decide about the possibility of offloading is focused on energy saving. This is the reason why it is crucial to have accurate enough models for execution and communication that permit running meaningful simulations. Simplicity is still, however, such an important factor.

3.2.6.1. Local execution energy model

A simple linear model to characterize the energy consumption at the MT is proposed in equation (3):

$$E_{local} = \gamma * bits \qquad (3)$$

The adjustment of the coefficient γ is based on a regression taking the measures provided by [9] and shown in the following Figure 3.

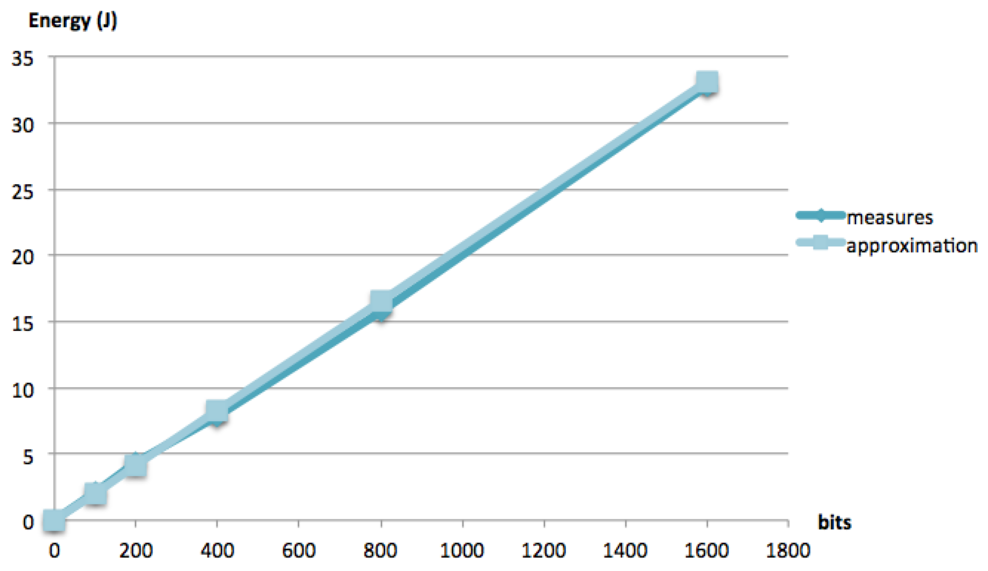


Figure 3: Local energy consumption linear model

In Figure 3, we observe the energy measured for different number of bits versus the linear approximation proposed to model the local execution energy model where the constant $\gamma = 20.66 * 10^{-3}\ (J/bit)$.

3.2.6.2. Communication energy model

The proposed model has been gathered from [10]. The expressions (4), (5), (6) and (7) have been isolated in a function where communication rates are provided (bits/s).

For the UL the power needed to transmit is calculated (4). The DL is also characterized with the power needed to receive (5). When the MT transmits, the power consumption in the RF (radio frequency) circuits increases with the radiated power. In addition to that, a baseline power is consumed just for having the transmitter chain switched on. On the other hand, the energy spent in the baseband processing circuits is also affected by the turbo encoding whose complexity depends on the UL data rate. Practical measurements available in the literature show that when putting all this together, the total UL power consumption is greatly affected by the radiated power but its dependence with respect to the encoding rate is not that important. Based on this, we will model the power required by the terminal for an UL transmission as a linear function of the transmission power plus a constant power that includes both the consumption for having the transmission circuitry on and also the consumption of the baseband circuits. Although this model is very simple, by adjusting properly the constants we can obtain a power estimation very close to the ones obtained with more complex models in 3GPP. But, using a simpler model will make easier to capture the essential trade-offs in our problem.

For the DL, when the MT is receiving the required power scales with the data rate. In addition to that, a baseline power is also consumed just for having the reception chain switched on. Although the receive power may affect in the AGC (Automatic Gain Control) circuits the dependency is not as significant as the rate.

The energy per transmitted bit is calculated with the rates given as a parameter of the function, for the UL (6) and for the DL (7).

$$p_{UL} \approx k_{tx,1} + k_{tx,2} * p_{tx} \text{ (Watts)} \qquad (4)$$

$$p_{DL} \approx k_{rx,1} + k_{rx,2} * r_{DL} \text{ (Watts)} \qquad (5)$$

$$e_{bit_{UL}} = \frac{p_{UL}}{r_{UL}} \text{ (J/b)} \qquad (6)$$

$$e_{bit_{DL}} = \frac{p_{DL}}{r_{DL}} \text{ (J/b)} \qquad (7)$$

Where p_{tx} represents the radiated power in UL, p_{UL} stands for the total power needed for the UL transmission (that encompasses the radiated power and the constant term $k_{tx,1}$ modelling the power needed to keep the terminal active). The constant $k_{tx,2}$ measures the linear increase of the transmitter power consumption with the radiated power.

p_{DL} stands for the total power needed for the DL transmission (that encompasses the power needed to receive signal and the constant term $k_{rx,1}$ representing the extra power consumption for having the reception circuitry switched on). $k_{rx,2}$ measures the increase of the power consumption with the decoding rate.

Parameters r_{UL} and r_{DL} are the transmission rates (b/s) for the UL and DL respectively used to perform the calculus of $e_{bit_{UL}}$ and $e_{bit_{DL}}$. These lasts are used to obtain the total energy used for communication e_{comm} in (8), where s_{UL} are the number of bits transmitted and s_{DL} are the number of bits received.

$$e_{comm} = s_{UL} * e_{bit_{UL}} + s_{DL} * e_{bit_{DL}} \qquad (8)$$

For each bit to process, it must be known how many bits have to be transmitted in the UL and received in the DL. This relation depends on the overhead generated by certain app and the communication system.

Formulas (4) and (5) are based on the following numerical values for the parameters related with energy consumption model: $k_{tx,1} = 0.4\,W$, $k_{tx,2} = 18$, $k_{rx,1} = 0.4W$, $k_{rx,2} = 2.86\,W/Mbps$. The maximum transmission power remains constant during the simulation: $p_{tx} = 100\,mW$. Additionally, the system does not concern about mobile state. The device never is off or in idle mode, that is, the terminal is always ready to transmit or receive with the consequences that entail in terms of energy consumption.

3.2.7. Other issues

APs have not been characterized as a resource themselves. Obviously, they are crucial resources in the scenario due to the fact that they route the load to the VM destiny. Nevertheless, routing is not taken into account in this project in the sense that we assume that they do not introduce routing delays. This does not mean this issue was not important enough. The reason why they were not included is because including routing issues would be a spark for future further research issues. In this occasion, the user just needs an AP to be connected to the network. There is no intention to manipulate realistic complex mesh nodes.

3.3. System modelling

This section describes a simplified guide to understand better the code of the project provided in my git repository [3]. Although the code is documented in the comments added, the main purpose is to explain briefly how it was built to understand better the code flow.

The simulator consists in an offloading manager that will control all the resources in the described scenario. The inputs by the user are:

- Number of MTs
- Numbers of APs
- Number of VMs
- $t_{threshold}$
- Number of bits in each packet
- Number of packets per second (λ)
- Simulation time
- Kind of application (index)

Despite this last long list, just the first three items are compulsory for running the simulator at first statement. The rest are just parameters inserted in the code but clearly identified so that they can be changed for running the tests.

Once the scenario is defined, the simulator creates the resource availability matrices for all the resources. There are the following resources:

- Radio access channel for each AP (UL and DL)
- Backhaul connection for each VM (forward and backwards)
- VMs occupation
- MT occupation (each terminal calculates the time to process a request taking into account its occupation so as to inform the manager)

The occupancies of the resources are expressed in terms of a collection of matrices associated, as shown in the following Figure 4.

$$\begin{pmatrix} t1 & t3 & \cdots \\ t2 & t4 & \end{pmatrix}$$

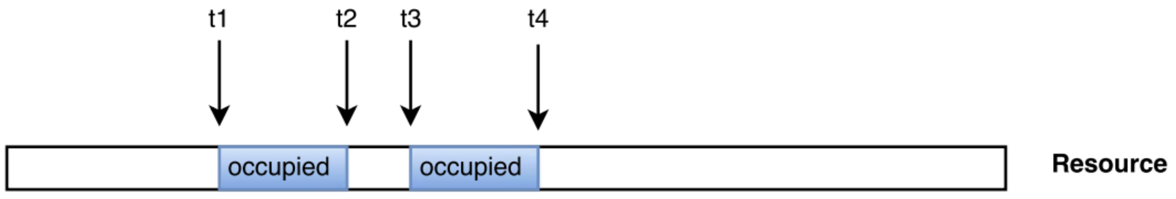


Figure 4: Resource time frame

This structure is repeated for all the resources listed previously. The resource occupation is managed by means of a matrix that marks the time frames when the resource is occupied, plotted as blue boxes in Figure 4. The first row of the matrix contains initial time of the occupation of the resource, whereas second row indicates final time for each occupancy.

In this context the manager can check the state of the resources that the user pretends to use for the different possibilities in each request. Therefore, it can calculate the time it would take offloading and check if it meets the time requirement.

Once the criterion is fulfilled, the manager updates the resources in use adding the time frame of the new occupation. If the request is performed after any other booked time gap, a new column is added, as shown in Figure 5. For the rest of cases, the manager appends the time used to the existing columns so as to minimize the numbers of columns. The aim is to have different columns just for disjoint time gaps. When the request is done before any booked time frame the new assignment will fusion time gaps enlarging the existing ones, not adding overlapped reservations (see Figure 6).

$$\begin{pmatrix} t1 & t3 & t5 & \cdots \\ t2 & t4 & t6 & \end{pmatrix}$$

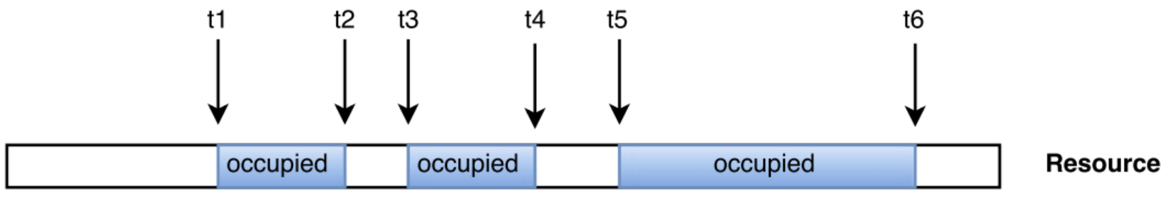


Figure 5: Basic new time frame addition

$$\begin{pmatrix} t1 & t4 \\ t2 & t6 \end{pmatrix}\!\cdots$$

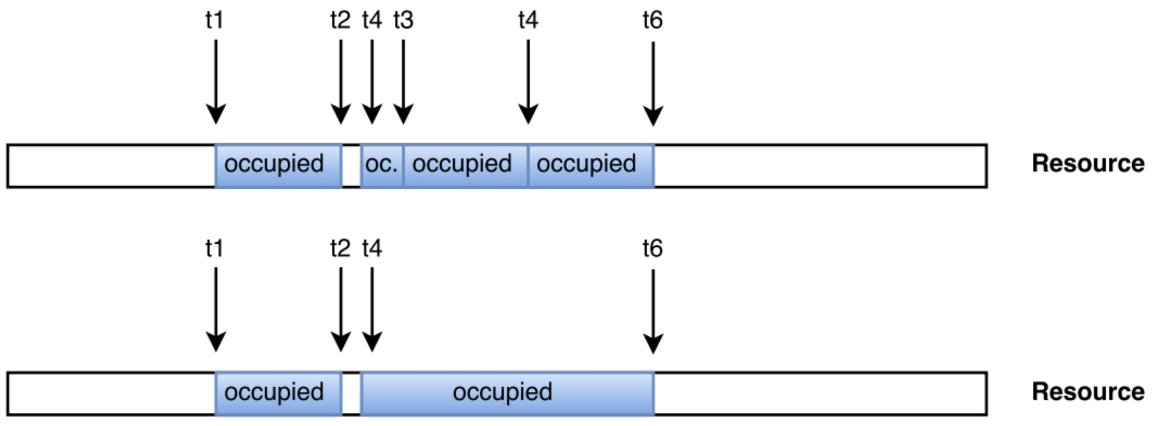


Figure 6: Time frames joint

With this mechanism the status of the resources is updated for each request. Consequently, the manager will know the occupation for the following request and, therefore, also the associated waiting times. The system does not care who is using what in each instant, it just worries about the occupation itself so it can check resource availability for future requests coming.

MTs control their occupation following the same methodology. For each new request, the terminal checks its availability so waiting times are also taken into account in the calculation of the processing time needed that will be crucial to set threshold time used by the manager. When local processing is the solution for the request, the resource assignment is equal to those exposed in Figure 5 or Figure 6.

3.4. Offloading management strategy

The offloading manager is the entity that will know the states of the resources all the time. Obviously, that could be feasible just in small networks, enough for the accomplishment of this project.

The offloading manager will distribute the available resources in the network for each request by any of the users. To achieve it, the offloading manager will follow the flow diagram detailed in Figure 7 where yellow boxes represent the actions performed by the MT.

According to Figure 7, suppose that a certain user generates a new request for processing b bits. On one hand, the manager needs to know the time that the terminal would need to process this packet of b bits. This time depends on how fast the terminal can process (*CPU*) and how busy it is (*state*). Once the manager knows this total time, a threshold time ($t_{threshold}$) can be set. Setting this value consists on deciding how exigent is the system in terms of time consumption. For instance, in this explanation $t_{threshold}$ will be set as the time that the MT would need to execute the process locally, also known as K=1 in (9).

$$t_{threshold} = K * t_{local} \qquad \text{with K>1} \qquad (9)$$

On the other hand, the energy that the terminal would need to execute the request in local has to be informed to the manager so that the manager can compare this information with the energy required to transmit and receive the corresponding bits in case that offloading was performed. The values for transmission and reception depend on the communication rates and the quantity of bits to communicate. With these two inputs, the system is ready to list the priorities in terms of energy consumption. This list contains all the possibilities the user have: all the possible options for offloading (i.e., all the possible combinations of APs and VMs) and local processing of the data at the MT. In case that the best options produce the same energy consumption, the most efficient option in terms of time will be used, obviously if it meets the requirement $t_{total} < t_{threshold}$, where the t_{total} is the total time it would takes to get the response.

Checking the resource availability consists in inspecting the state of the channel radio, the backhaul and the VMs. The manager figure out the total time needed for the offloading possibilities. This consists in taking a “photo” of the current state of the resources that may be used and checking the amount of time that would be spent in them for each possibility, taking also into account waiting in the queues. It must be said that we consider that all resources, allow packet fragmentation. This does not mean that packets can be processed by different VMs or routed by different APs. This fragmentation consists on the usage of the channel, the backhaul and the VMs. In that way, when the user tries to introduce the package in radio channel for example, it does not wait until there is a time gap big enough. It starts using free gaps until it could transmit the totality of the packet (see Figure 6).

After all these calculations, the manager can compare t_{total} with the threshold $t_{threshold}$ calculated previously. If it meets the requirement, then the resource can be assigned. If not, it must keep searching for another option. The system will stop looking for options when local option appears. This option will always have $t_{total} \leq t_{threshold}$ since in this project the threshold time is always greater or equal than local (9) since the aim is reducing energy consumption and not necessarily achieving faster processing. Nevertheless, in some occasions, the time of execution can also be reduced.

When the decision is taken, it either indicates the AP and VM that are now booked by the user who created the request or push the user to execute it locally at the MT.

Until the decision of offloading or not for this request is taken, any other request cannot be resolved. The system will always wait for new requests to start the decision cycle again.

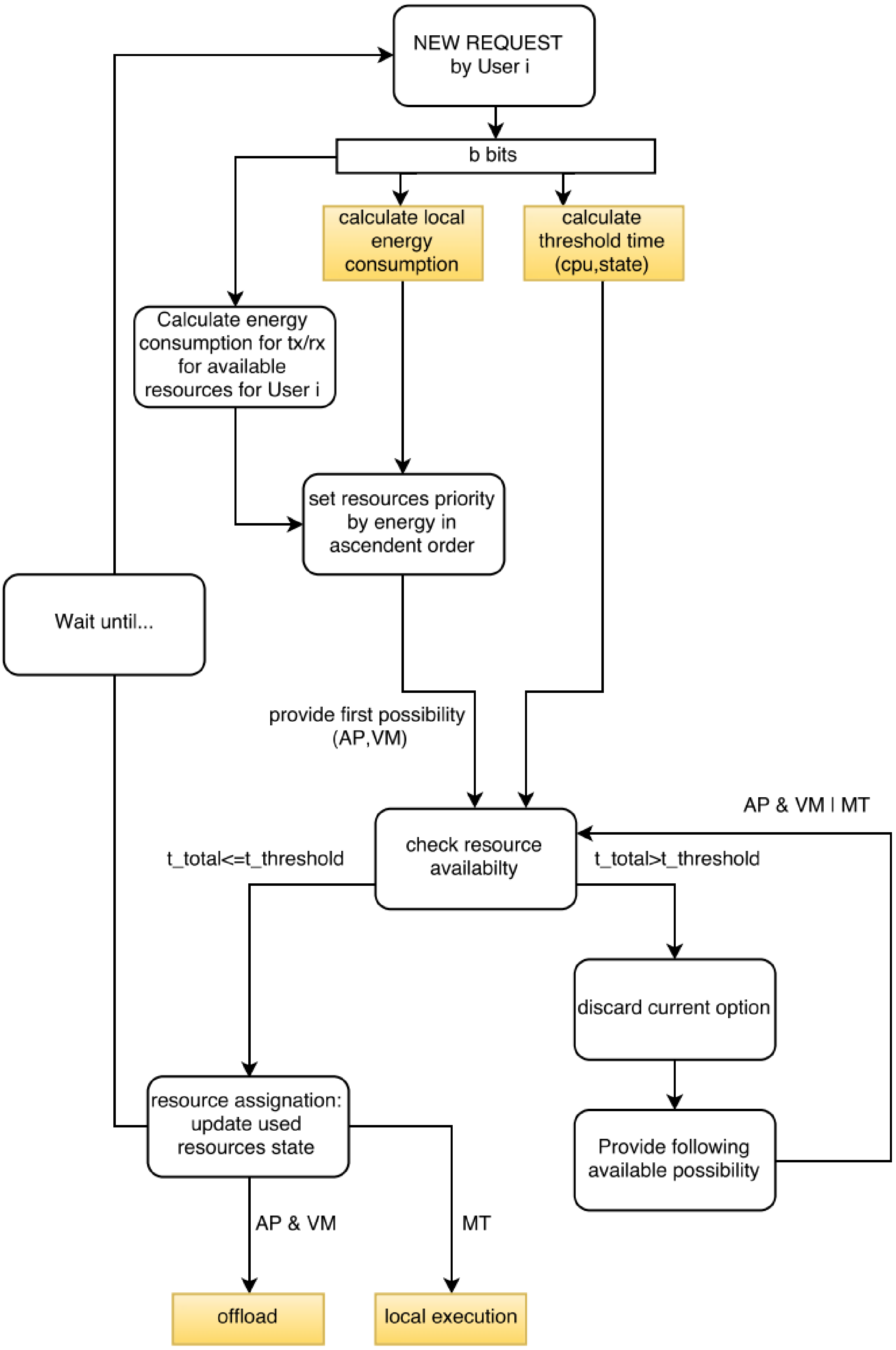

NEW REQUEST by User i
b bits
calculate local energy consumption
calculate threshold time (cpu,state)
Calculate energy consumption for tx/rx for available resources for User i
set resources priority by energy in ascendent order
Wait until...
provide first possibility (AP,VM)
AP & VM I MT
t_total<=t_threshold
check resource availabilty
t_total>t_threshold
discard current option
resource assignation: update used resources state
Provide following available possibility
AP & VM
MT
offload
local execution

Figure 7: Offloading manager strategy diagram

4. Results

This section reports the results obtained in the analysis period. Different tests are exposed, organized in different sections regarding their different purposes. However, each test case can contain different conclusions obtained from the same results.

There are five different sections where slightly different configurations allowed us to consider different scenarios in order to obtain different conclusions.

4.1. Local computational power importance

This test was done with the following input parameters: 35 MTs, 12 APs and 10 VMs.

The rates for UL and DL remain constant in the radio access channel and also the forward and backwards rates in the backhaul. As far as the computational power is concerned, the VMs power is the same for all of them, as well as its location, whereas the values for the MTs are intentionally different. The users will try to offload with the same communication rates.

The reason why this test is done with a high number of MTs, APs and VMs is to create a crowded environment where the channel is not always available so offloading will not always be an option. In that way, it will be possible to determine what users tend to offload more.

The algorithm defines the threshold time $t_{threshold}$ for each decision. This means that every time a user requests to offload a package, it will tell how is the terminal in terms of time availability so as to set the current threshold. The threshold is defined based on (9). It depends proportionally on the time that in each case the MT would need to process a certain request, taking also into account the state of the local resource. The value is over the time in local since it is just a restriction that allows certain delay and the system is trying to minimize the energy consumption as principal aim. In this particular test K=1.3, what means the criterion allows to exceed a maximum of 30% of the time the MT would need for processing the request locally.

MTs differ in the bit rate to process packages of bits. The size of the packages remains constant for the simulation as well as the kind of application. In other words, they would process the same number of bits in the same way but with different processing speeds.

The following Figure 8 is just one of the numerous executions realized for this test. All the results were pretty similar to this one, at least for the most meaningful concepts.

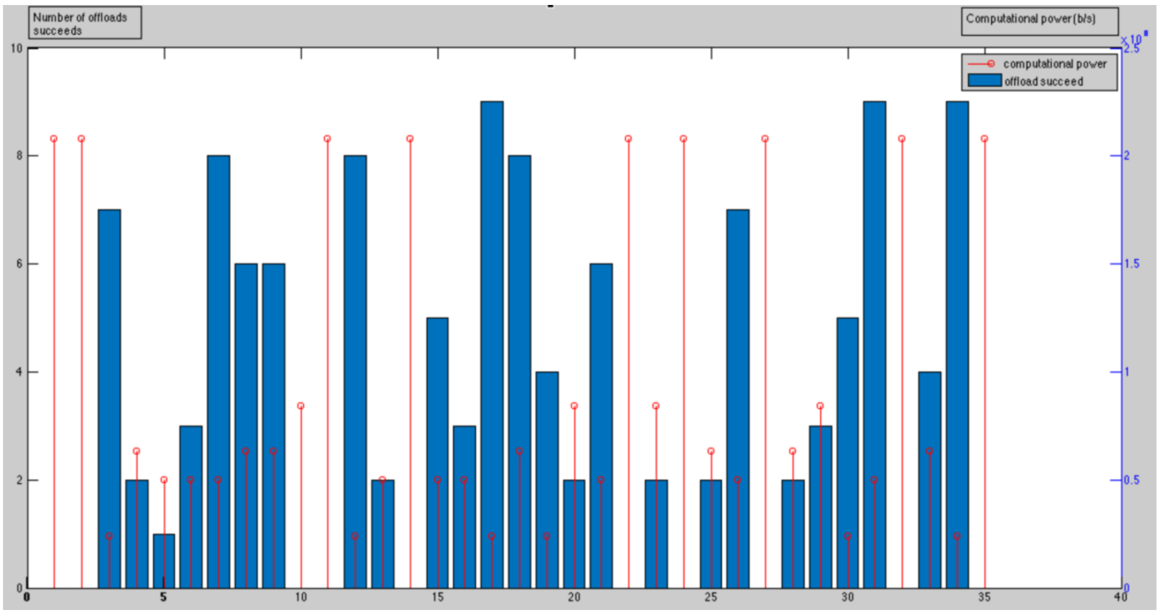


Figure 8: Number of offloads versus mobile computational power

In the left axis, associated to blue bars, we have the number of times that it is decided to offload for each of the 35 users producing requests. In the right one, the processing rates (bits/s) are represented again for each of the users included in the simulation. From the results, it is concluded that users who do not have powerful MTs are more likely to offload. It seems reasonable if we review the criterion to offload. The most powerful terminals demand a lower $t_{threshold}$ whereas for the least powerful MTs set $t_{threshold}$ much less demanding. Although the algorithm takes into account local state, the computational power results much more relevant when defining $t_{threshold}$.

After executing the same simulation several times, the values shown in Tables 8 were gathered to calculate in average the number of offloads done for each kind of terminal. For users who have a regular terminal, i.e., with an average calculation rate, the number of offloading success depends much more on the channel state than on the restrictions they set. This is the reason why some of the executions shown in Table 8 do not strictly follow the rule that less powerful devices are more successful in terms of offloading. Occupation issues introduce these slim differences. In fact, if we take a look again at Figure 8 we observe the same. The meaningful values are for the most powerful devices that do not offload due to their time thresholds whereas the ones who reach a higher number of offloads have weak devices. The devices with computational power between those are more sensitive to the channel and resources availability.

The kind of application defines the rates available for local processing depending on the complexity in terms of calculation for the terminal. This test scenario has been run several times and for two kinds of applications: App A and App B. The following Tables 8 show the average number of offloads performed by MTs characterized by their computational power (Mbits/s). Each isolated box corresponds to one single run. Particularly, there are four for each application. Both columns show that the behaviour is not different depending on the rates defining the kind of application in study.

App A	
Mbits/s	Avarage number of offloads
24	6.167
49.6	4.400
63.2	3.556
84	1.727
208	0.000
24	5.333
49.6	4.333
63.2	3.000
84	2.000
208	0.364
24	6.000
49.6	4.000
63.2	1.833
84	2.000
208	0.250
24	5.429
49.6	3.625
63.2	2.000
84	4.833
208	0.200

App B	
Mbits/s	Avarage number of offloads
37.6	6.857
76.8	2.333
97.6	3.429
129.6	2.200
320	0.375
37.6	5.571
76.8	3.400
97.6	4.500
129.6	2.714
320	0.333
37.6	6.250
76.8	4.083
97.6	2.700
129.6	4.000
320	0.000
37.6	4.933
76.8	2.250
97.6	1.400
129.6	0.750
320	0.000

Table 8: Average number of offloads simulation results

4.2. Remote machine distance and computational power relevance

The goal of this test was to detect the relevance of the distance between the AP and the VM versus the computational power of it. This project goes further than just using the computational power of VMs hosted in the APs. The main idea is to be able to wide the scenario where the data could be transmitted to farther computational entities. This implies the fact that the backhaul will be crucial to determine when offloading is adequate. In particular, this test relies on the latency that is introduced in the backhaul and how this has an impact on the system in terms of required computational power. This test has been run considering constant rates in radio channel to focus the study on the backhaul, the latency and computational power of remote VMs.

One MT will try to offload through a single available AP. There are two VMs: one located in the AP and a second one located in a farther cloud. The first option corresponds to offload whenever in the VM located in the AP, since radio channel is completely free and the consumption will be reduced. The second option is to go farther and process the request in a remote machine through the backhaul. If we consider this test was done in ETSETB university campus, Table 4 lists the places where the servers are hosted in this test. Obviously, all these values in Table 4 are just an approximation but at least they are indicative of how far the considered VMs are. In fact, it is a vast approximation since the routing and commutation are not specifically taken into consideration.

This test consisted in calculating theoretically the values exposed in Figure 9 and after verify the hypothesis with simulations. Figure 9 shows the relation between the distance and the computational power factor. This consists on comparing how fast the machine located in the AP processes the request, and how fast the remote machine must process the request to compensate the latency suffered in the backhaul (see (10) and (11)).

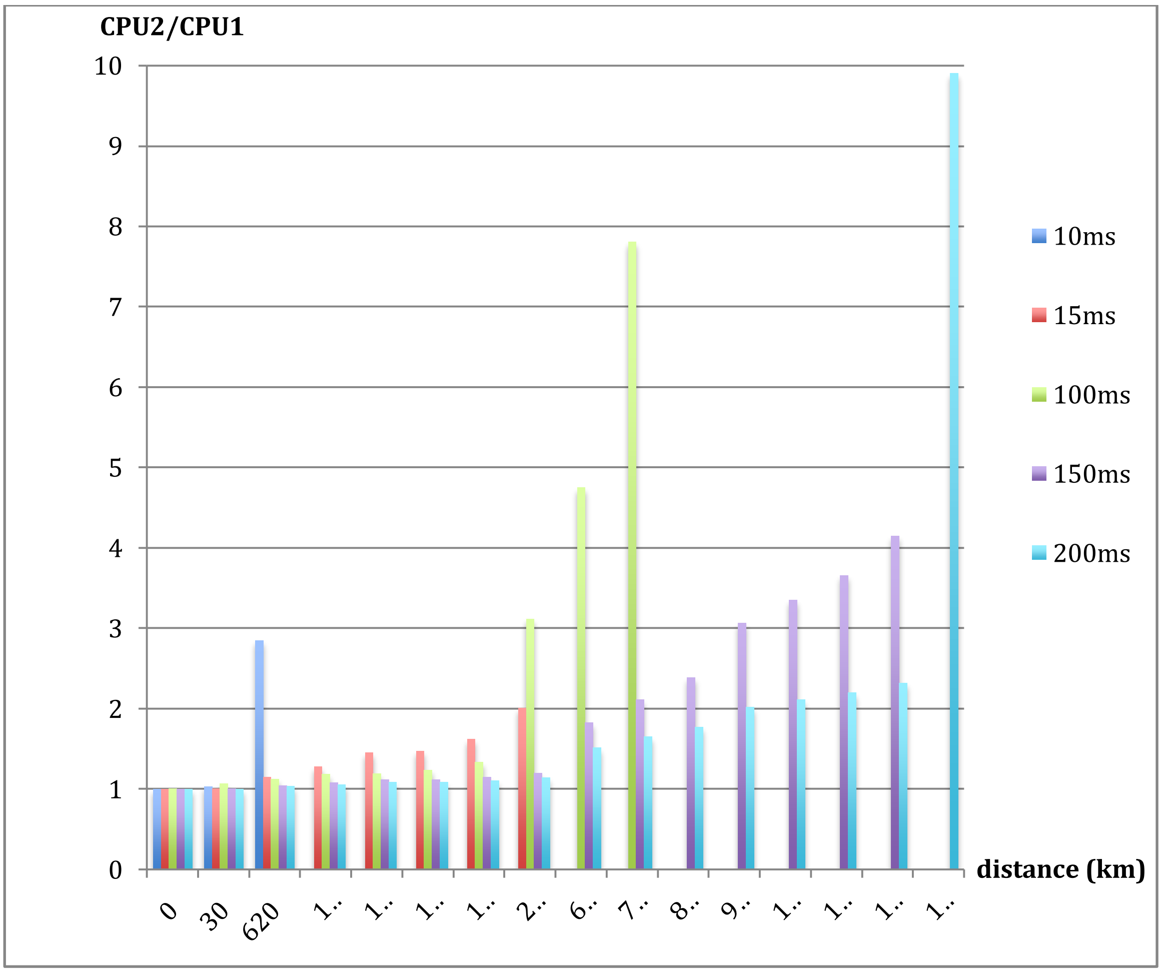


Figure 9: Computational power demanded for remote machines

$$ideal\ processing\ time\ in\ VM_{remote} = processing\ time\ in\ VM_{local} - latency \qquad (10)$$

$$CPU\ factor = \frac{processing\ \ time\ \ in\ VM_{local}}{ideal\ \ processing\ \ \ time\ \ in\ VM_{remote}} \qquad (11)$$

Formula (10) indicates that the processing time in the remote VM has to be equal to the processing time of the machine in the AP, with the extra requirement of latency compensation. The principal aim is to calculate how more powerful VM_{remote} must be to compensate the latency introduced due to its location. Therefore, and since VMs are physically distributed with different distances, the latency in (10) can be calculated using formula (2). In this particular case, the backhaul is using optical fibre: $n = 1.57 > 1$.

The VM physically located next to the AP (VM_{local}) is the reference since it does not suffer latency. For example, if we set that this machine would take 100ms to process certain request, the rest of ideal processing times would be based on based on (10). In figure 9, there are five different series. Each of them shows the seconds that VM_{local} will need to process the request by the user. CPU factors are the indicator of how much faster VMs must be in comparison to the one located next to the AP. The relation CPU_2/CPU_1 in Figure 9 is the factor obtained in (11) for each pair of local and remote VMs.

After analysing this theoretical study, some simulations had to be done. These simulations just consisted on doing a comparison of the occupancy of two VMs: always the one located in the AP and the different ones in the farther cloud. When assigning the computational power of the different remote VMs, the values exposed in Figure 9 were used. The fact that both options are equal in terms of energy consumption, since the communication rate in the radio channel is the same, the distribution of the requests depends on the time consumption. The best option in terms of time will be selected. Thanks to the difference in the computational power, VM_{remote} consumes the same time as VM_{local} . Consequently, the number of requests attended in one VM or the other one should be similar. Indeed, it is what tests results have shown.

Due to the fact that the results obtained in the simulation meet the theoretical hypothesis, there are different conclusions that might be recognized easily but deserve to be mentioned. This test remarks:

- Calculation time determines the maximum scenario radius

  The time needed for calculation defines the maximum distance that can be reached (i.e., the maximum distance at which a VM can be located so that it is decided that it will be used to process offloaded data). The results showed that the bigger the packages were the more distant a remote machine can be located. Not only the number of bits to process but also the complexity of the process determines the time spent in the remote machine. A complex process with low load can require a lot of time in the VM. Therefore, calculation time is what really defines how far the VM can be.

- Balanced load is not feasible for homogeneous networks

  The system should try to distribute the requests among the different available resources. In case that the radio channel states are constant and equal for all users, as in this test, the energy consumed to transmit and receive will be the same. Consequently, the different offloading possibilities have the same energy consumption cost and the criterion is to serve the one with the best time performance (always under the threshold time defined previously). In order to balance the load it is crucial to have similar requirements in terms of time. In the case of having a homogeneous network where all the remote machines are equal, this is not accomplished if some backhaul links introduce latency. The system will never deliver a request to the VMs located in a far location. This last will be under-used while the closest one, for example a VM residing at the AP, will be busy most of the time.

- Computational power for remote VMs

  Although in this test computational power specific values are not provided, CPU factors tell us the magnitude of the machines needed to be part of the scenario. The power required in maximum stage are reachable. Several configurations could compose a realistic scenario to use offloading as a solution for energy saving.

4.3. App complexity

In this case study, the relevance of the complexity of the process to run is tested. The app complexity is characterized with a factor denoted by α. This factor adjusts the time spent on the terminal processing the same amount of data (for example, an app with $\alpha = 2$ needs two times the time that an app with $\alpha = 1$ needs to process the same quantity of data). In order to do meaningful comparisons, the number of bits to process remains the same for all executions as well as the computational power of the terminals.

In this scenario, there are five MTs, one AP and two VMs. We assume that both VMs have the same computational power. The users generate new packages with the same rate and the same bulk of bits:

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$$
$$b_1 = b_2 = b_3 = b_4 = b_5$$

where λ_i is the package generation rate and b_i is the number of bits in each package for user i.

On the contrary, we assume that each MT make requests for running apps with different complexities modelled by means of different values of α_i, as follows:

$$\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < \alpha_5$$

User number five is the one who requests the execution of the most complex process, whereas user number one makes requests for running the lightest app.

The influence of this factor is directly related to the time that the user needs to process the bulk of bits. Consequently, since the time criterion is based on the time execution needed in local and the current occupation status in the terminal, the following results shown in Table 9 reveals that heavier processes lead to a more frequent usage of the offloading solution. This is reasonable since applications running complex processes set higher time restrictions values. Moreover, heavier processes energy consumption is higher too.

α	Offloaded served requests	Local execution
1	1.33%	98.67%
1.5	11.89%	88.11%
2	31.11%	68.89%
2.5	41.22%	58.78%
3	47.44%	52.56%

Table 9: App complexity influence on offloading technique simulation results

The results shown above are the average of several runs of different simulations in the same scenario. The simulator evaluates the number of times the user can offload so the repetition of these executions gives us the chance to calculate an average indicative value to describe the system behaviour.

Inevitably, in (9), t_{local} of complex apps enlarge the restriction so the offloading will take place more times. For this reason, we can conclude that the more complex apps are the more advantage of offloading the user will take.

4.4. Energy saving

In this scenario, there are five MTs, one AP and two VMs. One of the VMs is located at the AP, whereas the other one is at a remote position. The load will be distributed approximately equally among VMs due to the fact that a more powerful VM is assigned in the remote position. In this experiment the number of requests per user in a second will be the variable under observation. We will assume that the number of packets per second (λ_i) is the same for all the users in each simulation.

In this scenario configuration, two case studies are analysed. In both case studies the input parameters are the same. These cases are the following:

- Only local execution is possible.
- Local and offloading can be selected.

The simulator ran tests for different values of λ_i forcing, in the first case, that offloading could never be an option. These results correspond to complete local execution, which means that there is no energy saving at all thanks to offloading. The results are then compared to the other extreme situation (i.e., the second case), in which all the requests are handled through offloading, providing energy saving. To force the extreme situation of always offloading, the communication rates assigned were the highest possible.

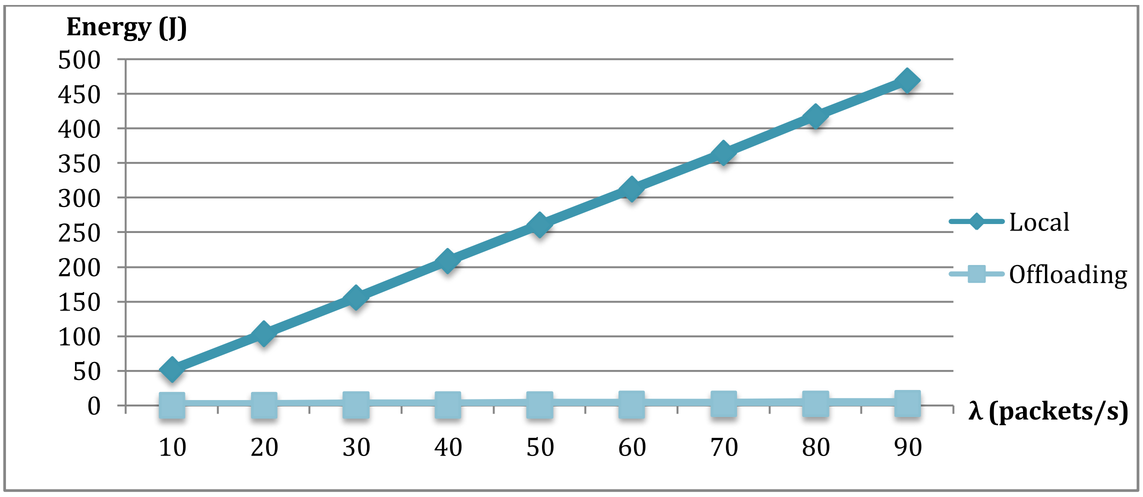


Figure 10: Local/Offloading energy expense for different packets per second

Local	
λ (packets/s)	Energy (J)
10	52.3
20	104.5
30	156.8
40	209.1
50	261.4
60	313.6
70	365.9
80	418.2
90	470.4

Offloading	
λ (packets/s)	Energy (J)
10	0.3
20	0.6
30	1.0
40	1.3
50	1.7
60	2.0
70	2.4
80	2.7
90	3.0

Table 10: Simulation results showing the Local/Offloading energy expense for different values of packets per second

The energy values gathered represent the accumulated energy consumed by one of the users in a simulation of ten seconds. In fact, the value is the same for all the users since all of them play under the same conditions.

Figure 10 represents graphically the results contained in Tables 10. Although there is of course energy consumption also for offloading, it is almost negligible in contrast to the case of local computation. This means that the energy required to transmit and receive data through the wireless channel is lower than the energy required to process such data at the MT. However, this astonishing result should not be misunderstood. Offloading would only offer such an energy saving if all the packets generated by the users could be offloaded, which is almost impossible in a realistic environment because not all the users will have the best communication rates possible during all the process, which is what we have forced in this concrete simulation setup.

The purpose of showing these two extreme cases is to quantify the maximum gain in terms of energy saving that offloading could provide in ideal scenarios. In a realistic deployment, the real energy saving is never that much, but the introduction of offloading makes possible to reduce the energy consumption approximately a factor 10^2 at maximum.

Notice the fact that the gap in terms of energy saving between local and offloading increases with λ_i. The consumption grows linearly with λ_i, as clearly seen in the plot shown in Figure 10. This means that processes that require processing a higher amount of packets take more advantage of offloading in terms of energy saving.

This test does not include plots where some packets are offloaded and some are executed locally. However, Figure 10 can help us to justify that using offloading for a percentage of the load will help to reduce the energy consumption of MTs. At this point we would like to emphasize that there are many elements that impact on the possibility of offloading. In general, MTs will always tend to offload as much as possible; however, many other variables that define the scenario may impact on the final decision.

Now, offloading plot will be inspected to describe its behaviour with respect to λ_i. Full offloading is possible when radio transmission rates through the wireless channel are good enough. Therefore, it is completely necessary providing excellent rates to force full offloading. In Figure 11 two different radio transmissions rates are assumed in order to characterize the scenario.

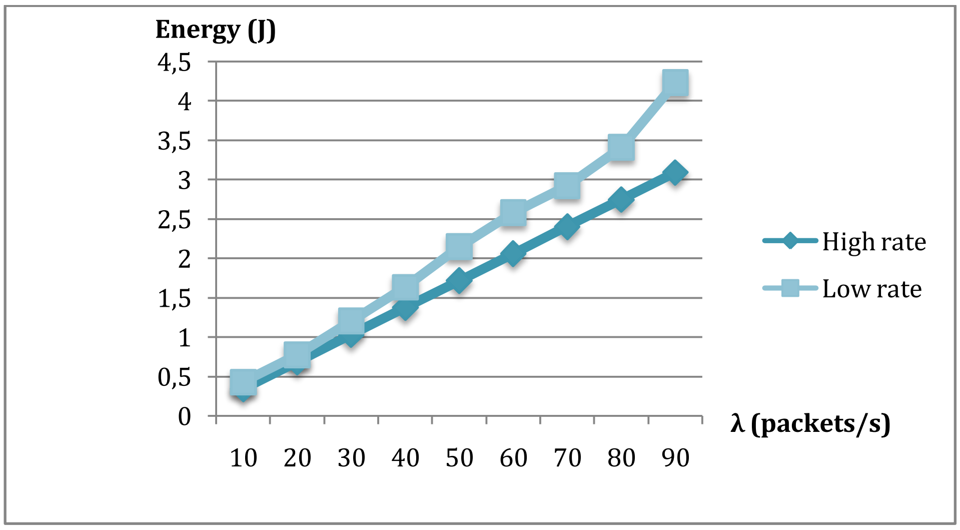


Figure 11: Energy expense for high/low communication rates for different values of packets per second

Offloading (High rate)	
λ (packets/s)	Energy (J)
10	0.3
20	0.6
30	1.0
40	1.3
50	1.7
60	2.0
70	2.4
80	2.7
90	3.0

Offloading (Low rate)	
λ (packets/s)	E (J)
10	0.4
20	0.7
30	1.2
40	1.6
50	2.1
60	2.6
70	2.9
80	3.4
90	4.2

Table 11: Simulation results concerning energy expense for high/low communication rate for different values of packets per second

In Table 11, on one hand the results on the left table correspond to the scenario where the assigned radio channel rates are the best possible (i.e., the highest MCS in Table 2). On the other hand, the results in the table on the right correspond to worse radio channel conditions (index=13 in Table 2). With a constant transmission power, the rate will primarily depend on the state of the channel. Nonetheless, both channel conditions are good enough to be able to offload a high percentage of packets. As it can be seen in the tables, the values of energy spending for the low rate are greater than for the high rate.

Anyhow, what these last results show us is the fact that the dependence of the energy with respect to λ is approximately linear. The greater λ_i is, the greater energy needed. Further than this, while carrying out this test, the results revealed that the radio channel rates have a direct impact on the performance. When the rates were worse, the number of offloads was critically reduced so energy saving was minor. Figure 11 shows very low values of energy consumption in both cases because both rates are relatively good in comparison with the ones that could be assigned in LTE (see Table 2). In a nutshell, the terminals that have better channel conditions and, therefore, can support higher communication rates, will take much more advantage of this technique in terms of energy saving.

4.5. Time restriction behaviour

Note that while the algorithm presented so far tries to minimize the energy consumption, time remains as the restriction. The key point of this test is to check how the time consumption behaves for all those cases.

For this test, it is not possible to obtain an average value that includes each single possible test run, since variables such as communication rates or computational power are not randomly fixed. There is no need to define a particular scenario. In fact, the variety of scenarios resulted fruitful to observe that the behaviour in discussion was not fair just for certain configurations. The only value to be remarked in this test is the $t_{threshold}$ that is a

30% higher than the time that the MT would need to run the process, taking into account the occupancy of the processor in the MT in each request (t_{local}).

In this section one single result is exposed of the all collection of experiments done. Each of the bars in Figure 12 shows us the percentage of time used for the offloading procedure, considering 0% the time t_{local} . Surprisingly there are many of them below the time that the device would need in local. In other words, in several of these cases the system allows to reduce also the time consumption thanks to offloading. Notice that the rest of them are below the threshold set (in fact, it is the reason why they accessed to offload).

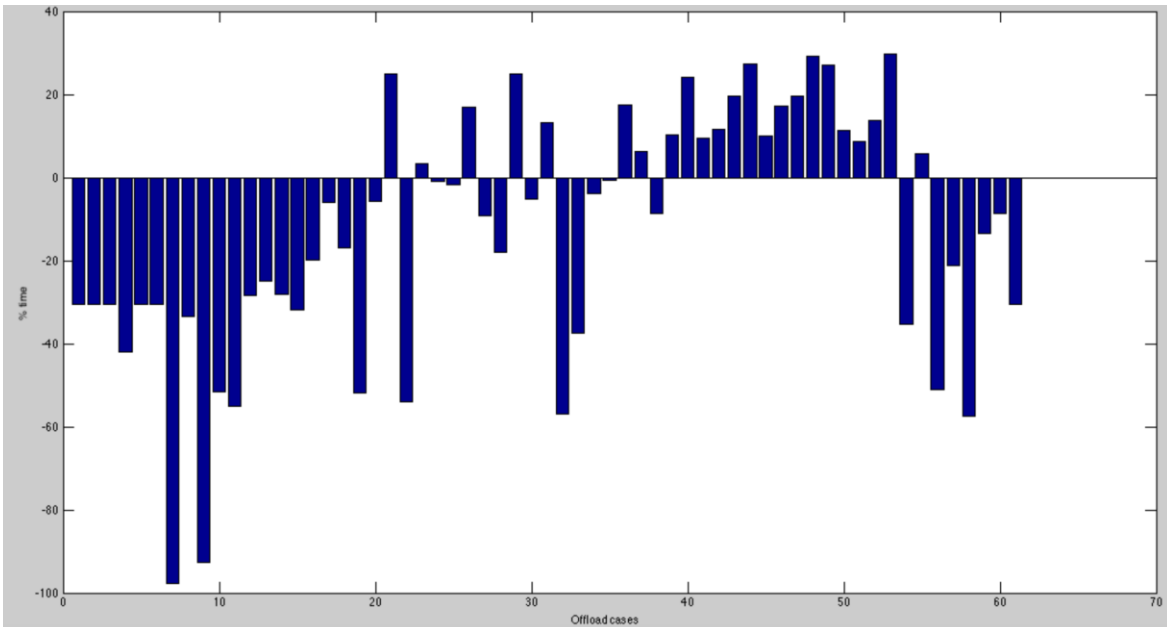


Figure 12: Threshold time accomplishment

It is important to remark that the target of the algorithm is not to reduce the time needed to run processes but to reduce the energy consumption. Logically, if we can even improve in terms of time consumptions, that is good news for the user who will simultaneously save energy and reduce the time needed to obtain the result. Obtaining such hopeful results tell us that the restriction could be less narrow than the one selected for the test.

This opens a new door to applications that do not allow delays, buts this is out of the scope of the project. Nevertheless, the results of this test are encouraging because the possibility is right there, something that a priori was discarded.

5. Budget

The elaboration of this thesis has consisted basically in the development of a simulator as it has been pointed out during the entire document. Therefore, to create this economical budget we will only have to take into account the following items: the license of the software (Matlab in this case) and the human work of a junior engineer. The final cost analysis is shown in Table 12.

Task	Weeks	Hours	Cost/hour (€)	Cost (€)
Scenario definition	3	30	8	240
Design strategy	7	154	8	1,232
Simulator development	9	198	8	1,584
Evaluation and analysis	3	66	8	528
Redaction	4	40	8	320
Software license	-	-	-	1,000
TOTAL				4,904

Table 12: Budget of the project

6. Conclusions and future development

The motivation of this project was to prove the power of offloading so as to provide the user a longer battery-life cycle. Fortunately, the project has shown what desired: offloading as an energy saving strategy is feasible. Along the different tests exposed, the users that offloaded could save energy.

One of the most relevant issues relies on the radio communication rates. In all the tests, the users who were unfortunate in the rates assignation never could take advantage of offloading. The communication rate defines the time spent in wireless communications, therefore it defines the difference of the energy cost for either processing or communicating. When the rates are low, they involve greater energy per bit values so this technique could not be seen as an energy saving strategy. Since the importance of communications rates has been experimentally verified, the study of different wireless communications technologies, such as WLAN or GPRS, will remain in the backlog for future research.

The rest of variables that have been pointed out are time spending descriptors: the number of bits, the complexity of the app and the local computational power. All of them influence on the time needed for local processing at the MT. The longer the process is, the more energy the terminal needs. With the proposed strategy, where the decision is taken depending on local energy consumption, all of these variables are implicitly defining the energy consumed in local. All these variables remarked that the right configuration to use offloading to save energy is the one that contemplates long time executions in local. It could be provoked because the bulk of bits are big, the complexity of the app is high or because local computational power is not good enough to hold certain processes. Furthermore, a long execution enables offloading since the time restriction set will not be too restrictive. In fact, we particularly verified that users holding weaker terminals tend to offload more whereas users with powerful devices were obstinate to use the technique because their exigencies were much more restrictive.

On the other side, the machines serving the users have also been under study. In this occasion, the relevance for the user is how fast these machines could serve their requests so as to accomplish the time requirement. The experiments showed us that the idea of offloading to clouds with machines located in farther locations than the AP is feasible. The machines just have to be powerful enough to compensate the latencies associated to the transmission through the backhaul links. Nowadays, with the building of machines clusters, CPU requirements are completely reasonable and can be provided. Nevertheless, in this project, the networks were simplistic and routing was not included in the scope of the study. Further research must be done in this field so as to sharpen the results and the exigencies of serving remote machines.

Surprisingly, the energy saving strategy proposal could also be used for mobile applications that do not allow time delays in the execution. Indeed, this kind of application was not contemplated since the project goal was just to reduce the average energy consumption and time was in the background. The results showed that even for the most exigent applications offloading could be a solution to expand battery lifetime.

Bibliography

[1] SPCOM group, TROPIC: Distributed computing, storage and radio resource allocation over cooperative femtocells", 2015

[2] Dinh, H. T., Lee, C., Niyato, D. and Wang, P, "A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Communications in Mobile Computing" , 2013

[3] M.Lucchini, "My personal TFG repository" (https://github.com/maurolucc/Offloading), 2015

[4] R.Agustí, F.Bernardo, F.Casadevall, R.Ferrús, J.Pérez, O.Sallent, "LTE:Nuevas tendencias en comunicaciones móviles", 2010

[5] 3GPP, "LTE; Requirements for further advancements for Evolved Universal Terrestrial Radio Access (E-UTRA) (LTE-Advanced)", (3GPP TR 36.913 version 12.1.0 Release 12), 2013

[6] Primate labs, "Samsung Galaxy S6 edge", 2015

[7] Passmark Software, "Benchmark test description", 2015

[8] Passmark Software, "CPU Benchmark", 2015

[9] Milindkumar H. Tandel, Vijay S. Venkitachalam, "Cloud Computing in Smartphone: Is offloading a better-bet?*", 2014*

[10] O.Muñoz, A.Pascual and J.Vidal,"Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading", 2014

Glossary

AP:	Access point
CC:	Cloud Computing
CPU:	Central processing unit
DL:	Downlink
GPRS:	General Packet Radio Service
LTE:	Long term evolution
MCC:	Mobile Cloud Computing
MCS:	Modulation Coding Scheme
MT:	Mobile terminal
QoS:	Quality of Service
PRB:	Physical Resource Block
SC:	Small cell
SNR:	Signal Noise Ratio
TDMA:	Time Division Multiple Access
UL:	Uplink
VM:	Virtual machine
WIMAX:	Worldwide Interoperability for Microwave Access
WLAN:	Wireless Local Area Network