ANNÉE 2017

THÈSE / UNIVERSITÉ DE RENNES 1

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MATISSE

présentée par

Ismael Cuadrado-Cordero

préparée à l'unité de recherche n^o 6074 - IRISA
Institut de Recherche en Informatique et Systèmes Aléatoires
ISTIC

Microclouds: An Approach for a Network-Aware Energy-Efficient Decentralized Cloud

Thèse soutenue à Rennes le 9 fèvrier 2017

devant le jury composé de :

Guillaume Pierre
Professeur, Université Rennes 1/ president

Pierre Sens
Professeur, Université Paris 6 / rapporteur

Patricia Stolf
Maître de Conférences HDR, Université de Toulouse Jean Jaurs, / rapporteur

Guillaume Urvoy-Keller
Professeur, Université Nice Sophia Antipolis / examinateur

Adrien Lebre
Chargé de recherche Inria (en détachement de Mines Nantes) / examinateur

Chris Phillips
Maître de Conférences, Queen Mary University of London / examinateur

Christine Morin
Directrice de recherche, INRIA / directrice de thèse

Anne-Cécile Orgerie
Chargée de recherche, CNRS / co-directrice de thèse

Acknowledgements

We are formed out of those people we share our experiences with. As such, this thesis is the result of three years of work, but also a life of experiences and people who have lead me to be the professional I am.

First, I would like to convey my sincere gratitude to the members of my jury, who have been interested on my work and have taken the time to evaluate it. I thank Patricia Stolf and Pierre Sens for accepting to review my work, and for their thorough comments, as well as their encouraging support. I also want to thank Adrien Lebre, Guillaume Urvoy-Keller and Chris Phillips, for their interest in my work and for taking the time and effort to attend to my dissertation. Finally, I thank Guillaume Pierre for accepting to be the president of my jury.

Credit should also be given to my advisors, whose support and guidance have been decisive to the fulfillment of this thesis. Christine, I thank you for welcoming me in Myriads, it has been a privilege being part of this team. Also, I thank you for your patience and judicious advise, which I believe made me a better scientist. To Anne-Cécile, I thank you for your guidance and for believing in me all these years. You have not only shown me the way, but you have always provided the means so I could continually challenge myself. I am proud of being your first PhD student :-).

During all these years, Irisa has been like a second home to me. To all of you that were part of it during my thesis, I want to share my warmest thanks. After all these years you have been many the people who have shared with me many social breaks. I am grateful to all of you, for the multiple activities, the warm debates, thorough comments on my work and personal support in everything I have done. Special thanks are due to David and Yunbo. I could have not found better office mates. I will greatly miss an unmatched work environment. I would not like to forget all those coworkers who have left Irisa. To Alexandra, who was a great support during my first year in France, and to Mar, who always gave the right advise, I thank you.

To the networks team of the Queen Mary University of London I also owe a praise. To Felix, Ignacio, Amna, Timm and Gareth my warmest thanks. You really made me feel at home and all I learned from you helped me developing microcities.

Along all my life, I have found some pillars which have brought the best of me. The first one has been sports. Thus, it could not let pass the opportunity to acknowledge all of you who have challenged me to balance my mind and body. In this section I would like to thank Eric and Raul, who have both physically challenged me to reach my best, each in his own field.

The second pillar of my life have always been my friends. As they are too many to enumerate here, I want you all to know that I carry you on my heart. While writing this thesis I had the invaluable support of Bogdan and Madalina. You both will always have a place in my heart. Also, I have relied on Younes at many times. For your all your support, thanks. To Luismi, for he showed me the world of research. You advised me during both my bachelor and masters' thesis. This PhD thesis is also the result of all your advise. To Rocio, whose support during these three years should not be forgotten. Finally, Fernando has always been there to debate. We might not always agree on everything, but you have taught me to listen to other points of view, no matter how ludicrous they seem.

A third pillar on my life have been my family. I am grateful for how lucky I was, and I know that I would have never achieved this without your endless support, your education and your love. You have always believed in me, and you have always been there for me. Special thanks to my mother, for your strength and for your selflessness, which are an example to all of us. To my father, for your joy of life and courage. To my sister, for being fearless and for standing up for yourself. To my cousin, for a very thoughtful video, and to all its participants. To Genc, for you have won this place by your own merits.

Finally, the last pillar on my life is sustained by a single person. To Anna, I want to express my deepest love and gratitude. A thesis is never easy, but you have always been there in the bad moments. You know how to make me smile and you have always believed in me. Without you I would have never succeeded. It takes a very special type of person to do all that you have done for me during these years. I love you.

To all of you, thank you. This thesis belongs to you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Cloud Computing is an Internet-based technology that provides pooled resources as a business model. It offers on-demand, flexible services adapted to the needs of the clients, making this technology more accessible to a wide range of users who cannot afford maintaining a large computational power, which is often underutilized [4, 5]. Cloud Computing has given these users the opportunity of renting a service during a period of time, reducing costs and increasing their productivity [6, 7].

Cloud Computing is nowadays widely spread, reaching, according to recent studies, its maturity phase as a technology [8, 9]. In the last years, the adoption of Cloud technologies has ramped both in the number of users and companies [10–12]. According to the same studies, users and companies found that, using Cloud Computing, the need for over-provisioning their resources has been reduced.

To supply users the needed resources, Cloud providers build datacenters which host the necessary hardware. To use a cloud service, clients connect to the datacenter through Wide Area Network (WAN) connections such as the Internet. This approach has been fundamental for the adoption of the Cloud, since the operation of large datacenters decreased the costs of electricity, network bandwidth, operations, software and hardware in computing [5]. Notwithstanding, the datacenters-centric architecture also suffers downsides. Some downsides are delays in communication between final users and scalability issues related with the physical constraints of datacenters resources [13]. According to the same study, in order to host the growing demand in Cloud, computing datacenters should exponentially upgrade their computational, network and storage resources. These upgrades need to be addressed through a modification of current infrastructures.

Scaling the resources of a datacenter to a point where they cannot be hosted in their current physical infrastructure - that is, it is needed to either increase its size or create a new datacenter to host the new resources - is an expensive and complicated process [14]. Moreover, datacenters are energy-intensive facilities, which pose a threat on the environment [15–17]. Furthermore, the number of technological solutions and users is projected to boom. Indeed, according to Cisco [18], more than 500 million devices where added to the network only in 2015. With the expected increase of data produced by these new connections, the global energy consumption is expected to increase. Indeed, according to Andrae and Edler [19], 66% of the total energy consumption of communications technologies will be used by datacenters and fixed area networks by 2030.

On the one hand, the growth in number of users' is directly related with the overturning in Cloud's energy efficiency. This increase and decentralization of users causes infrastructure providers such as datacenter operators and Internet Service Providers (ISPs) to over-provision their resources to host the ever-growing demand and traffic. Over-sizing an infrastructure implies that some resources are idle most of the time, but they need to stay active and use energy in case of a potential increase in demand [20, 21]. On the other hand, the dispersion of users across different locations forces ISPs to upgrade their infrastructures, to manage a larger amount of data, and establish new connections in remote areas [22].

Since 2010, companies have worked on reducing their energy consumption while increasing the number of datacenters, as shown by Shehabi et al. [23]. The reduction on the increase of energy consumption by datacenters shown in this study shows a turn towards energy efficiency. However, recent studies in datacenters' energy consumption [24, 25] show that the energy efficiency (ratio between energy needed and consumed) provided by Cloud datacenters has found a turning point. That is, datacenters have a worse average Power Usage Effectiveness (PUE - a unit that measures how much of the energy used by a facility is wasted) than it was in the beginning of this decade, according to the same studies.

The Cloud's datacenter-centralized architecture forces other actors involved in the communication (such as Internet Service Providers) to oversize their infrastructure in consequence [26]. In a datacenter-centralized architecture, when two users need to communicate, data are sent along the Internet to a datacenter, which is not always the shortest path. Keeping all the needed networking resources available affects the total energy consumption of the Cloud. Under the expected increase in the number of Cloud Computing users, nowadays infrastructures will need to be upgraded to ensure the provision of the same Quality-of-Service to the user. Furthermore, they will need to be upgraded to minimize the impact in energy consumption [27]. The increase in devices contributing to the Cloud creates new research challenges in networks [28].

1.2 Research problem and goal

The estimated energy usage of the whole ICT ecosystem is around 10% of all the energy production [29]. Of these, datacenters and core networks - main network connecting points of presence of network providers - adds up to 2-4% of the global energy production [30,31]. Still, only 42.3% of the world's population is estimated to be connected [32]. According to this study, in the period 2000-2014, the number of Internet users has increased tenfold, with an accentuated growth between 2010 and 2014.

From a networking point of view, the uptake of mobile and Cloud technologies has increased the global IP traffic five times between 2009 and 2014 [33]. According to Cisco, the global IP traffic is envisioned to increase threefold over the next five years [33]. In fact, according to Cisco [34], mobile wireless traffic will exceed wired traffic by 2018. Furthermore, by 2020 the number of devices connected to the Internet will be as much as 3 times the global population, according to the same study. The consequential growing distribution of users adds to the limitations of mobile devices in terms of resources and connectivity [35], bringing new challenges to the Cloud both in terms of energy consumption and Quality of Service.

This current and expected growth in number and distribution of new users is meant to be a critical issue in the Internet in the near future, due to several reasons. On the one hand, the addition of new connected devices is forcing ISPs' network to increase their resources, which are limited to the available raw material and space inside their own facilities; and it increases their energy consumption per network [36]. On the other hand, the computation that a datacenter can host is limited to the resources they can accommodate. As a consequence, the number of Cloud datacenters in the world keeps growing in the last years [11] (additionally to Dell'Oro Group, server shipments for Cloud datacenters surpassed a third of total shipments in the first quarter of 2016 [37]). As for the ISPs, this situation is equally increasing both the electricity bill of these companies and the global energy consumption.

As a consequence of the increase in the number and distribution of users, companies changed their view of Cloud Computing, splitting their resources in different locations to provide a closer service than centralizing services in only one datacenter [38]. For example, Amazon has passed from having all its datacenters located in the region of Northern Virginia (US) in 2006 to being present in 13 different geographical regions in 2016, expecting to add 4 new geographic regions by 2017 [38]. In 2016, European Amazon users can place their Virtual Machines in datacenters in 2 locations inside Europe: Ireland and Germany.

The multiple datacenters architecture provides a service more suitable to the user and reduces latency [39]. Yet, it still suffers from downsides. In these applications where data are not widely spread [40, 41], the distance traveled by data is far larger than necessary.

An example of locally accessed information can be found in Wikipedia. This online encyclopedia counted in August 2016, 292 different language editions, most of them accessed locally by users of a specific community, although it used a multiple datacenters' Cloud approach. For example, one of this languages distributions is Breton, a Celtic language spoken in the French region of Brittany. A user accessing Wikipedia's Breton distribution from Brittany will need, at least, 10 hops (or level-3 network devices) before entering the Wikipedia "Breton" servers, physically located in Ireland. After entering the internal network, the numbers of internal hops is only known by the company.

In this example, data travels further than necessary for a mainly regional utilization, changing networks (when changing between countries). This distance in communications affects the user, who perceives a higher delay [42].

Besides, with the evolution of connected devices, the amount of data sent through the network grows. For example, after the adoption of smart-city initiatives all around the world (Santa Cruz, Amsterdam, Barcelona, etc.) [43], the load of data in networks increases. In smart-cities, local authorities deploy interconnected platforms of hardware which provides useful information for their citizens through wireless networks used to provide services such as reducing road traffic or managing emergency situations. Since this information is used by the citizens of the city, data can be stored close to the clients. If the smart-city application uses a datacenter-centralized architecture to store its data, ISPs need to keep a reachable path between both clients and the datacenters, providing enough bandwidth to transfer these extra data. This causes the ISPs' energy consumption to increase. ISPs can adapt their infrastructures according to the traffic across their networks. Thus, if the traffic is lower, ISPs can downgrade their infrastructures, and ultimately the network devices, to function at a lower performance and save energy [44, 45].

Moreover, centralizing data in datacenters located in a foreign country introduces a large and unnecessary delay in communications. In the case of latency-critical applications, the delay in communications caused by the distance traveled by the data from the clients to the datacenter may be unacceptable [46, 47].

The main goal of this thesis is to reduce the energy consumption of the Cloud, while providing a better Quality-of-Service to the user. My work is focused on Cloud applications with low propagation of data. As studied by Hristova et al. [40] and Chat et al. [41], in many cases data are not propagated far from the source, that is, in many cases several users using an application are geographically close to each other. While different Cloud applications exist, we have decided to focus on those that provide simultaneous access to multiple users. In these applications, all the users concurrently work on the same data, and delays in the update of information affects the experience of the user [48]. Our main targets are energy utilization of Cloud applications and Quality-of-Service offered to the users, focusing on the latency in these scenarios where the computation and data can be split into smaller chunks.

On the one hand, distributed Clouds address the problem of latency. In a distributed Cloud, the computation is run on several devices, closer to the clients [49]. Distributed Clouds achieve a lower latency by replicating content along multiple devices (including the clients devices, in some approaches). While distributed Clouds have been deeply studied in parallel computation and content distribution, they do not fit the problem defined before. In the first place, the replication of data and computation of distributed approaches makes them unsuitable for interactive applications with multiple contributors, as replication is either causing consistency problems between different versions or requires a high speed communication between nodes [50, 51]. Moreover, the replication increases the broadband utilization of the network due to the extra data sent to synchronize different versions [52]. As a consequence, the energy consumption in the ISPs' networks is also increased, given that their network resources need to work in a higher performance mode to successfully route the increased traffic in the network.

On the other hand, some approaches rely on a unique manager to delegate the computation among different nodes. These approaches are called distributed [53, 54]. However, existing distributed approaches still lean on datacenters, normally the ISPs' [55, 56]. They are usually infrastructure-dependent (that is, they make use of specific infrastructures) and do not consider the client machine as part of the network. While this solution reduces the latency in communications compared to datacenter-centralized ones, it only translates the problems of datacenter-centralization from the Cloud provider to the ISP. However, data remains, in many cases, far from the user, and the energy expenditure and latency could be improved. Furthermore, in some approaches, the datacenter acts as a cache, keeping a copy of the data that synchronizes with the datacenter periodically. Other distributed approaches, such as desktop computing [57, 58] consider heterogeneous devices (such as personal computers) to host distributed computation. However, these approaches are designed to serve computation to the manager, but do not communicate between them [59].

None of the existing approaches offers a solution which provides low latency and low energy consumption for dynamic applications where several users interact with data in real-time, leveraging from the computational power of all elements in the network. Moreover, modern distributed and distributed approaches consider heterogeneous devices as part of their infrastructure, and some nodes cannot contribute resources, while others (including state-of-the-art network devices) can contribute greatly. This situation needs also to be addressed, as for an architecture to be accepted it should be considered fair between its users.

The challenge taken in this work is to provide an energy-efficient Cloud solution which provides a better Quality-of-Service than existing approaches, without depending on a specific infrastructure. On the one hand, we believe that the use of an approach which centralizes computation in datacenters becomes counterproductive from an energy-efficient perspective. On the other hand, the use of distributed approaches becomes inefficient from the perspective of the network provider, as they make an excessive use of the network, due to replication [60]. Furthermore, the ISPs' network which interconnects the final user and the datacenter, has been neglected in the literature, and reducing the traffic along a network is proved to save energy [61, 62].

1.3 Contributions

I will demonstrate that energy consumption can be drastically reduced through the use of a distributed approach, while providing a better Quality of Service for the user than centralized approaches. Moreover, I will show that a significant share of the data sent through to the Cloud can be processed locally, which reduces energy consumption and increases the Quality of Service to the user. To locally manage and host this computation, we propose *GRaNADA*, a distributed PaaS system which geographically distributes the computation between the devices involved in the communication (such as client and network devices) to save energy and provide a better QoS. *GRaNADA* does not eliminate the need for centralized datacenters, but withdraws from them all computation prone to be localized, to be processed closer to the service end users. Along with *GRaNADA*, we propose *DEEPACC*, a Cloud-aware routing protocol which plans a distribution of the connections between the nodes based on energy-efficiency and Quality-of-Service of the system. Using our solution, services are hosted in the involved devices (either the clients or network devices with application layers). The sum of all involved devices form an overlay platform called *microcloud*.

In our work, we apply GRaNADA into 2 different contexts. In the first one, we aim to reduce the energy consumption of Cloud systems paying special attention to the network, as it has been overlooked in previous works. This first context adapts GRaNADA for its deployment in a large scale system, such as a core network. To evaluate our system, we built a prototype of GRaNADA and simulated it on different networks, using NS3 [63], a packet-level network simulator. The use of a packet level simulator allows us to obtain accurate measures of a real network, using different approaches, without having to interfere with the active network devices. On the first set of experiments, we simulated a core network and compared our solution to different approaches. We show that this approach is able to save up to 75% of the energy compared with a caching approach, and 10 times less than the usual centralized Cloud architecture. This contribution is presented in Chapter 4 and has been the subject of a publication in the 11th IEEE International Conference on Green Computing (GreenCom2015) [64]:

> Ismael Cuadrado Cordero, Anne-Cécile Orgerie and Christine Morin, "GRaNADA: A Network-Aware and Energy-Efficient PaaS Cloud Architecture", *GreenCom: IEEE International Conference on Green Computing and Communications*, Sydney, Australia, pages 412-419, December 2015.

Secondly, we apply GRaNADA in a more localized context. In this context, the need for centralized datacenters is completely obliterated and all the computation are located in a densely populated but small area, such as a neighborhood. Through the use of GRaNADA we propose a real-life platform which enhances the smart-city paradigm, that we called *microcities*. In this scenario, we leverage the unused resources from both the users and smart-city network's devices to host services targeting a specific neighborhood. Once again, to evaluate our system, we adapted the existing prototype of GRaNADA and simulated it on different networks, using NS3 [63]. We show that, constraining the computation to a small geographical area through the use of *microclouds*, the user experiences a much better Quality of Service (QoS) than using datacenter-centralized Cloud services. This contribution is presented in Chapter 5 and has been the subject of a publication in the

16th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP2016) [65]

> Ismael Cuadrado Cordero, Felix Cuadrado, Chris Phillips, Anne-Cécile Orgerie and Christine Morin, "Microcities: a Platform based on Microclouds for Neighborhood Services", *ICA3PP: International Conference on Algorithms and Architectures for Parallel Processing*, Granada, Spain, pages 192-202, December 2016.

Finally, we acknowledge that users might be reluctant to adopt a new technology where some of them contribute more resources than others (or not at all). Thus, to stimulate the adoption of our technology, we propose as our last contribution the creation of a market of services between the users, so that users are remunerated for the utilization of their resources. To do so, we have opted for an approach based on multiple auctions. In our approach, we consider the application provider as a party interested in the result of the auction (the application provider might be interested in, for example, assuring a good QoS for every user to obtain certain popularity), and we decided to include it one in the auction process. We propose an auction system adapted to the *microcities* paradigm, which we called *micromarkets*. We show that, using our system, users perceive a service more suitable to their needs, enhancing their *satisfaction* - difference between the requested and received QoS. On the other hand, users leasing their resources are compensated for their services, while service owners - the persons or companies in charge of the application - attract more users. This contribution is presented in Chapter 6 and has been the subject of a publication in the 5th IEEE International Conference on Cloud Networks (CloudNet2016) [66]:

> Ismael Cuadrado Cordero, Anne-Cécile Orgerie and Christine Morin, "Incentives for Mobile Cloud Environments through P2P Auctions", *CloudNet: IEEE International Conference on Cloud Networking*, Pisa, Italy, pages 248-253, October 2016.

1.4 Organization of the Manuscript

The reminder of this thesis is structured as follows. In Chapter 2, we show an evaluation of different existing centralized and decentralized approaches and a state of the art on network-aware and Quality of Service approaches. In Chapter 3, we describe the concept of *microclouds* and a prototype of its software architecture (*GRaNADA* and *DEEPACC*). In Chapter 4 we apply GRaNADA into core networks, and evaluate its suitability focusing in its energy consumption, comparing it to a distributed and datacenter-centralized approaches. In Chapter 5 we apply GRaNADA in a smart-city context where the users are located in localized geographical areas and we evaluate its suitability from the point of view of the Quality of Service. In Chapter 6 we design a market of services' system to incentive the utilization of microclouds; and we propose and evaluate a new approach based on double auctions, *micromarkets*. Finally, in Chapter 7 we draw the conclusions of the current thesis and propose future research directions.

Chapter 2

State of the Art

This chapter provides the necessary background for the contributions of this PhD thesis. First, we present the existing Cloud infrastructures and a taxonomy of applications and architectures. Then, we address the main requirements in terms of Quality of Service to which the Cloud is subject. Finally, we present the energy consumption aspects of Cloud Computing and existing literature on energy efficiency improvements in Cloud computing.

2.1 Introduction

In Section 2.1.1, we provide definitions of important notions in Cloud computing. Once the main terms have been defined, we present a taxonomy of existing Clouds in Section 2.1.2 and a taxonomy of Cloud applications which are relevant for our work in Section 2.1.3.

2.1.1 Cloud Computing Definition

The U.S. National Institute for Standards and Technology (NIST) [67] defines Cloud computing as "a computation model designed for enabling *on demand network* access to a convenient *shared pool of configurable computing resources.*". These resources must be *rapidly provisioned* and released with minimal management effort [67].

The entity which serves the Cloud computing services rented by the user is called the *service provider*. The service provider relies upon the use of virtual environments, such as *Virtual Machines (VMs)*, to provide the computing resources. A VM is a software emulation of a computer architecture and operating system, which is placed on top of the real hardware and allows the creation, destruction, migration and pause of the service. Through the use of Virtual Machines, the service provider makes a flexible and efficient use of hardware resources [68]. A *user's environment* is formed by one or more Virtual Machines.

The NIST defines the following requirements for a Cloud computing environment [67].

On demand activation: For the creation of new environments, no human interaction is necessary from the service provider's side. Until a Virtual Machine is used, it will not be created. In the same way, after a Virtual Machine has completed its purpose it is deleted. Finally, during periods in which the Virtual Machine is not used it can be saved and restored again when necessary. Through these mechanisms, the resources allocated by the Virtual Machines can be freed and used by others.

Resource pooling: The resources available for the Cloud are shared by all the users in the Cloud. Every resource available in the environment is introduced in a pool. During the creation of a Virtual Machine, the necessary resources are extracted from the pool and associated to the VM. When the Virtual Machine does not need the resources, these are put again in the pool.

Rapid elasticity: Addition or deletion of resources in a user's environment are performed at runtime. To the user, the resources appear to be unlimited and can be appropriated in any quantity at any time.

Measured service: The Cloud provider ensures that users are not over-provided nor under-provided either. Elasticity allows the provision of a "pay-as-you-go" payment model. Users pay only for the assigned resources. When needing a different number resources, the user extends the contract. Then, the Cloud provider meets the new demand and adapts the billing to the new amount of resources assigned to the user. To adapt the billing to the demand, the Cloud provider should have a metering system. The measuring and granularity of the metering used for the Cloud environment vary. The NIST uses the following metering characteristics as examples [67]:

- *Storage:* How much space the user is assigned. Users may pay different prices for different kinds of storage (e.g. Dropbox [69]).
- *Processing:* How much computational power the user has access to (e.g. Amazon EC2 [70] allows the user to select a configuration of memory, CPU, instance storage and partition size).
- *Bandwidth:* Users access to networking resources (e.g. Microsoft Azure [71] pricing on data transfers).

Multi-tenancy: From the previous requirements extracted, it can be extrapolated the multi-tenancy requirement. Cloud computing enables sharing of resources and costs. Clients share costs through the centralization of infrastructure. Instead of paying for the whole price of renting one or more servers, users pay only for the shares they are using (pay as you go). On the other hand, Cloud providers also benefit from sharing of resources. When users share resources in an infrastructure, Cloud providers can host more users at the same time than if each user was renting the entire resources but utilizing only part of it. Cloud providers profit from the utilization of datacenters that, if not, would be often only 10 to 20% utilized [72–74]. For example, Cloud providers can host several Virtual Machines in the same server. This way, unused servers can be freed and shut down, which saves energy and reduces the services provider's electricity bill.

Some other requirements can be implied from the previous definition. For instance, Dillon, Wu and Chang [75] determined that *broad network access* was also a Cloud requirement. This requirement ensures that a Cloud can be accessed from many heterogeneous devices and locations.

Also, *reliability and security* are main concerns in Cloud systems. Given their use in different contexts, such as business transactions, reliability is an important factor in Cloud computing. Due to the virtual nature of Cloud computing, the client can dynamically create a redundant and dependable environment. Ghobadi et al. [76] define the following issues, which Cloud computing should address.

- **Fault tolerance:** Disruption on application and/or backup should be transparent to the user.
- **Security:** In Cloud computing, the location of the data may vary in time, due to reallocation of VMs. Security of the service, such as confidentiality and privacy, should be always provided independently of the physical location.

- **Load balancing:** All services should be monitored to have knowledge of their traffic.
- **Interoperability:** Applications should be able to be ported between Clouds.

2.1.2 Classification of Clouds

Clouds are usually classified in two different schema, depending on their characteristics [67]. The first schema is called the *service model*. In this classification the client is the key factor, separating every Cloud system according to the knowledge or use that the client will make of it. The second classification is done according to the *deployment model*. In this schema the Cloud itself becomes the main factor for the classification, separating the different types of Cloud according to the implementation model that the Cloud requires.

Service Model Classification

The service model classification defines the following types of Cloud systems:

1. **Infrastructure as a Service (IaaS):** It provides an on demand infrastructure to the customer. A Cloud computing infrastructure, typically in a datacenter, is utilized. On top of it, a virtual environment in deployed. The software management and deployment of platforms and applications rely on the client. Development and/or installation of new programs and software management and configurations are controlled by the client. An example is found in Amazon EC2 [77]. Amazon offers computational power and storage and network infrastructures as part of its IaaS model. However, maintenance of the VMs is managed by the client.

2. **Platform as a Service (PaaS):** A PaaS Cloud offers a platform on which the client can develop applications. A PaaS deployment includes all the needed libraries and compilation environments. Programming and/or deployment and maintenance of Cloud applications are the main responsibilities of the client. On the other hand, the maintenance of both the infrastructure and the platform is done by the Cloud Provider. An example of a PaaS Cloud is Google App Engine [78].

3. **Software as a Service (SaaS):** SaaS Cloud is mostly focused in the exploitation of applications. In this Cloud, clients do not interact with the programming, deployment or maintenance of an application, neither the maintenance of platform or infrastructure. The Cloud provider manages the application for the client. The client may access the Cloud from different devices, and many companies develop different versions of their software for all devices. Google Drive [79], Gmail [80] and Microsoft SkyDrive [81] are examples of SaaS Clouds.

The previously described separation is shown in a layered structure, from lesser IT knowledge to more specialized knowledge, in Figure 2.1.

Deployment Model Classification

The second classification model for Clouds considers implementation of the Cloud, both for deployment and maintenance. This model is called *deployment model classification*. The deployment model classification provides the following classes, as defined by the NIST [67]: Private, Public and hybrid Clouds.

Private Cloud: A private Cloud is designed, maintained and used for and by a company solely. Sometimes, the Cloud may be operated by a Cloud provider, as long as

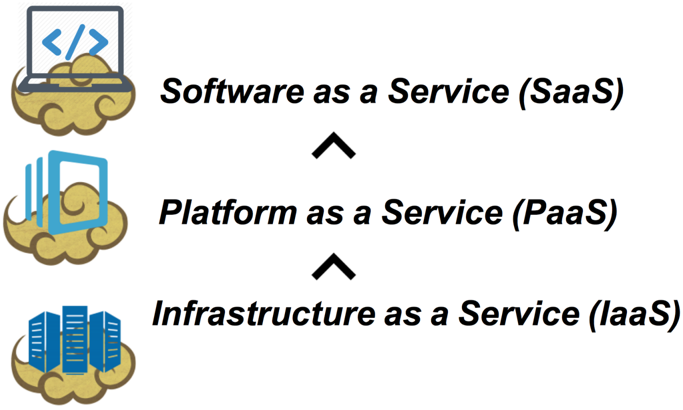


Figure 2.1: Service Models

a confidentiality contract exists between the Cloud provider and the given organization. Its utilization stays exclusively within the company itself, so it is not accessed by external parties. This type of Cloud provides control over the company's sensitive information, such as management of clients and employees, financial applications and project management. For example, IBM offers private Cloud computing services through SoftLayer [82].

Public Cloud: A public Cloud is hosted and controlled by a private company, but accessed by a wide range of clients for the use of it, independently from their company affiliation. Amazon EC2 provides public Cloud computing services [70]. If the infrastructure of a public Cloud is formed by more than one datacenter, a facility used to house multiple servers and computing components such as networking devices, then it is called a *multi-datacenter Cloud*.

Hybrid Cloud: The concept of hybrid Cloud focuses on enhancing the resources of a specific Cloud. In a hybrid Cloud, a private Cloud is extended by adding external resources from a public Cloud. The clients perceive the resulting Cloud as a single one, but both the private and public Clouds remain unique entities that work independently. Nowadays, public Cloud providers such as Google [83] or Microsoft [84] propose hybrid Clouds as part of their Cloud plans. Also, Amazon proposes *Virtual Private Cloud* [85] as a hybrid Cloud implementation. Here, a secure bridge between private and one or more public Clouds is implemented. This way, both the private and the public Clouds behave as one. The maintenance and deployment of the public Cloud is done by the public Cloud provider, while the private Cloud is maintained by the company itself. According to RightScale [9], the hybrid approach is the preferred by companies, with 80% of surveyed companies using a hybrid approach, compared to 74% in 2014. *Community or federated Clouds* are formed by different Clouds. These Clouds are perceived as a unique Cloud by the clients', similarly to hybrid Clouds. In contrast to to hybrid Clouds, federated Clouds establish a bridge between two or more companies' Clouds [86]. Another example of federated Clouds is found in Contrail Project [87].In this project, a group of independent Clouds is combined to create a federated one. The goal of this project is to deliver a Cloud software stack of components designed to work together in creating a federated Cloud.

Despite the general requirements and classification, Cloud computing is used to host different types of applications with different requirements. In Section 2.1.3, we classify most relevant types of applications that run on the Cloud. We compare these types of applications based on their requirements.

2.1.3 Classification of Cloud Applications

The "pay as you go" model allows application developers to create new applications with a more reduced prior investment. The reduction of preliminary costs has been welcomed by the independent software vendors [88]. The appearance of new software vendors implied the development of new application models. For example, Cloud computing has supported the development of new applications offering on-line services such as data editing or hosting. In this section we we provide a classification of applications running on Cloud infrastructures. This classification is especially relevant to understanding the different Cloud architectures addressed in Section 2.2, as different applications are suitable for different architectures.

An application is perceived by the clients as an instance. An instance of an application is a working environment which serves the same information to the clients. An instance of an application can be implemented in a single virtual process (e.g., creating a lightweight virtual container [89]) or several instances can be hosted in the same VM. We have considered the following types of applications in this work.

Document/Data editing: These applications allow one or more clients to concurrently modify a file. The share file can imply that clients share large amounts of data that are heavy on the network such as video files [90], photo albums [91], etc. or lighter files such as documents [81, 92] or audio tracks [93]. We have decided to consider both kinds of applications as the same type due to their shared characteristics, i.e. both applications are based on a simultaneous modification of a file by one or more clients. Instances of these applications count their concurrent clients in the tens (e.g. document shared in a team) or hundreds (e.g. an on-line form).

We can safely assume that in the case of editing light on-line documents there are no relevant variations in the amount of data sent by users in time (peak of data). On the other hand, it is possible that a variable number of users is connected to the document at the same time (peak of users). In the case of editing on-line files which are heavy on the network, peaks of data are a possibility. A file can also become popular, thus having a peak in connections (users).

Similarly, the volume of data between the users and the application (i.e. the amount of packets sent between the client and the application) is, by definition, light in the case of light on-line documents. In the case of data editing, it becomes heavier. The flow of data (i.e. the continuity needed by a connection) is continuous in all types of on-line editing applications, i.e. the connection between the client and the server remains open during the life of the instance.

Finally, latency tolerance varies between instances. For example, an instance where there is no concurrent modification of data can tolerate latency of several seconds. When a single user is modifying a file, the modifications can be stored locally and shared with some delay with the server without affecting the experience of the user. On the other hand, when more than one user concurrently modify the same file, latency becomes an issue. On-line editing requires users to receive updates in real time to provide a good Quality of Experience to the user, i.e. the user is satisfied with the service. Thus, we consider on-line editing as real-time systems. As stated by Hatley and Pirbhai [94], these systems need response times of milliseconds.

Data sharing/Storage: Data sharing applications distribute an immutable file among different clients. The most known example of data sharing applications are Context Delivery Networks (CDN) [95, 96] and Peer to Peer (P2P) networks [97]. Storage applications are similar to data sharing applications. A known example storage application is Dropbox [69].

Both storage and data sharing applications host data off the user's device which can be accessed by one or more users. In both applications, the flow of data is discrete, i.e. the connection can be broken and established again without affecting the experience of the user. Then, users can establish several different connections to retrieve a file. Also, users expect certain delay while recovering the data in both types of application. Some seconds of difference in downloading a file does not affect the experience of the user. The main difference between storage and data sharing applications is their main usage.

We classify as storage type all the applications which have a low number of clients (hundreds). In storage applications all users are known to the original file owner. For example, Dropbox offers plans for individuals or teams [98]. We can deduce that an instance has, most of the time, less than a thousand users. The offered plan for large teams assumes users are not dispersed in space (enterprise plan). Thus, less replication is needed to store the information than to spread it between many clients. Depending on the type of service provided by the application, these applications may host either few clients per instance (personal use) or thousands (corporative use). Servers of this application receive flows of data which vary depending on the file.

On the other hand, data sharing applications have more clients per instance. In the case of most popular files, the number of simultaneous users can reach hundreds of thousands. Figure 2.2a shows the amount of users concurrently using a single popular file, according to a specialized media. Also, in many cases users are geographically distributed. Figure 2.2b shows the distribution of users downloading a similar popular file. Then, data sharing applications may be heavier on the network if the file becomes popular.

TRACKER NAME	STATUS	SEEDERS	LEECHERS
udp://tracker.openbittorrent.com:80/announce	active	98967	31305
http://tracker.ex.ua/announce	active	88630	26537
udp://tracker.publicbt.com:80/announce	active	118698	35709
udp://pow7.com:80/announce	active	122598	36736
udp://open.demonii.com:1337/announce	active	163496	43558
udp://tracker.istole.it:80/announce	active	118025	43223

(a) Most popular torrents (source: [99])

#	Country	%	City	%
1	Australia	11.6%	Melbourne	3.2%
2	United States	9.3%	Athens	2.9%
3	United Kingdom	5.8%	Sydney	2.0%
4	Canada	5.2%	London	1.9%
5	The Netherlands	4.7%	Stockholm	1.7%
6	Philippines	4.6%	Amsterdam	1.7%
7	India	4.2%	Madrid	1.5%
8	Greece	3.6%	Warsaw	1.4%
9	Poland	3.0%	Brisbane	1.4%
10	Sweden	2.7%	Perth	1.3%

(b) Geographical distribution of users downloading the same file (source: [100])

Figure 2.2: Torrent users and distribution

Social Networks: In social network applications there is a set of clients which contribute data, and a set of clients which consume these data. This is different from data sharing applications, where the relation is one contributor and several consumers. Also, we consider that any application in which the data generation is many to many, e.g. RSS feeders [101] or Facebook [102], is a social network. Social networks can be implemented using a distributed or a centralized architecture.

Another difference with data sharing applications are instances of the application. In this type of application, users configure which contributors they want to retrieve data from. For example, in Facebook a user can configure from which other users to receive data from. In the Facebook example, few users have the exact same set of friends. Even, in the case in which two users have the exact same set of contributors, the retrieved data may be affected by other factors (e.g. Facebook's algorithm). Then, in social network applications an instance is used by a limited amount of users (normally a single one). This implies that many instances of applications need to be created to provide the service (as many as users).

On the other hand, social networks have a high internal communication between instances. When a contributor updates information, this information is sent to all the instances interested in this update. Then, users receive their new update. To receive the update, users establish short connections with the server to retrieve data in different moments. Latency is better tolerated in the retrieval of data, i.e. a latency of several seconds in the update of new information does not affect the Quality of Service to the user. We can safely assume that, similar to data sharing applications, some popular files may host thousands of clients.

Gaming: Gaming applications host interactive video-games where several clients compete in real time [103]. Due to their own nature, instances of these applications host few clients (rarely over ten). This type of application requires a low latency in communications and opens a continuous flow of information between different clients.

Streaming: Unlike gaming, in streaming applications there is no communication between users. In a streaming instance, a continuous flow of data is open from the creator to one or more clients. In a streaming application there is no reciprocal communication between clients, only from the creator of content to the consumers of content. Thus, a streaming application accepts latency. As long as the flow of data is continuous, the Quality of Experience of the user is not affected if there are several seconds between the creation of the context and its reception. An example of streaming applications is found in video streaming like YouTube [104] and Netflix [105]. In the case of popular files, they can host thousands or millions of clients per instance. For example, according to YouTube [106], the number of concurrent users in a same instance can be up to 2 millions. Streaming applications can be implemented using a distributed or a centralized architecture.

Scientific Computation: These applications perform resource intensive computation. They need to be allocated in an infrastructure able to provide enough computational power to fulfill their requirements. An example of this type of application is found in scientific parallel computation [107]. The devices involved in a scientific application should be physically close, in the same datacenter, to reduce the delay in communications. Instances of these applications normally counts with a low number of clients (normally one). There can be a heavy communication rate between instances, for example in parallel computation.

Table 2.1 shows a classification of the different Cloud applications described before and their main characteristics for each category.

	Peaks		Internal Flow Volume	External Flow Volume			Continuity	Latency
	Data	Clients		High	Medium	Low		
Document Editing		X	LOW			X	CONTINUOUS	N
Data Editing	X	X	MEDIUM	X	X		CONTINUOUS	N
Scientific Computation	X		HIGH	X	X		DISCRETE	N
Storage	X	X	LOW	X	X	X	DISCRETE	Y
Gaming		X	HIGH	X			CONTINUOUS	N
Data Share	X	X	MEDIUM	X	X		DISCRETE	Y
Social Networks		X	LOW			X	DISCRETE	Y
Streaming	X	X	MEDIUM	X	X		CONTINUOUS	N

Table 2.1: Classification of Cloud Applications

We focus on 5 characteristics to compare the different applications. These are: the variability in time of users accessing an instance and data sent to an instance of an application (i.e. the existence of peaks); the average volume of communication between instances (i.e. the amount of data on the network for the communication between instances of the application); the average volume of communication between the client and server (i.e. the average amount of data in the network for external communications); the continuity of the connection (i.e. if the connection can be broken without affecting the experience of the user); and tolerance to latency. These characteristics have been chosen because they affect both the load in the network and in the server. Load in the network and the server may affect the experience of the user if they are not considered beforehand.

First, we consider the variability on the number of expected connections and/or data sent to the server. We called these *peaks*. A peak is an outlier increase in the demand of a service. We distinguish between peaks of data and peaks of clients. If an instance of an application suffers peaks of data, it means that this instance receives more data than average. If the instance is not prepared for a dynamic load, the user may perceive interruptions in the use of the service. If an instance of an application receives more connections than average, then it is called a peak of clients. This situation may be managed through elasticity (i.e. assigning more resources to the instance). Calheiros et al. [108] use elasticity of the Cloud to dynamically adapt to workload changes and guarantee a good experience of utilization to clients. A thorough survey on elasticity in Cloud systems, addressing definitions, metrics and existing solutions has been published by Coutinho et al. [109].

The second important characteristic is the communication between instances (*internal communication*). Communication can be low (under 500 Mbps), medium (between 500 and 1000 Mbps) or high (over 1000Mbps), based on the options offered by Amazon EC2 [70]. If the internal communication is high, then the instances should be physically located in close hosts, to reduce latency [39]. Also, an unexpected high flow of data between instances may put pressure on the network. Then, the application may use all the internal network resources which have been assigned to it by the Cloud provider. This may cause the instances not to communicate correctly between them. A flow of data between instances higher than what the infrastructure can handle may affect the users' experience.

Some cloud applications can be implemented using a distributed or a centralized architecture, as explained in Section 2.2. In the centralized architecture data are kept in a single location, while in the distributed architecture data are stored in several locations.

Cloud applications show differences in the amount of data between instances according to the architecture used to be implemented. Due to replication, deployments of of Cloud applications using distributed architectures need a higher internal communication.

Similarly, an application can make a heavy use of the external network (i.e. the Internet). We called this the *external flow of data* (between the client and the instance). A high flow of data between clients and instances is also heavy on the Cloud provider's network. The application needs a negotiated bandwidth with the exterior which fits its needs. If not, the network may become a bottleneck. In this case, the experience of the user can be affected as described by Jensen et al. [110].

The application should also take into account the necessary continuity in the users' connections. If the network exceeds its capacity, clients may see their connections delayed or interrupted [111]. Some applications require a *continuous flow* of data between the client and the server. For example, gaming applications need a continuously open connection. On the other hand, some applications can open and close several connections without affecting the use that the client makes of the application. This is the case of data sharing applications. For example, BitTorrent [97] opens several connection to the same file (i.e. different instances which contain the same data) to speed up the recovery of the data. An application requiring a continuous flow of information is more sensitive to network problems.

Finally, *Latency tolerance* is also an important characteristic to consider in an application. We define latency as the interval of time between the request and the delivery of data. In some applications, latency should be as small as possible not to affect the experience of the user [48]. We call these applications which need a very small latency (in the order of tens of milliseconds) real-time applications. This category includes gaming applications, but also on-line shared documents and data editing applications. On the other hand, other applications can tolerate a higher latency than real-time applications. For example, in the case of distribution of social networks information a latency of several seconds is negligible [112, 113].

Depending on the type of architecture, an application may fit better one Cloud or another. For example, scientific computation applications need to be deployed on Clouds with a large computational power and low latency between instances. On the other hand, scientific computation applications accept a higher latency between the client and the server than other applications. In Section 2.2 we review the various architectures for Cloud computing.

2.2 Cloud Architectures

The architecture of a Cloud determines the structure and behavior of the different components in the Cloud. We distinguish between centralized and distributed Clouds. The infrastructure of a centralized Cloud is localized in a specific geographical area. In opposition to centralized architectures, a distributed Cloud is formed by different elements based in multiple distant locations. All these elements work together to create a single Cloud to the user over a Wide Area Network.

In this section we compare central and distributed architectures. We also evaluate how different architectures consider the stakeholders, in particular the Cloud provider (which manages the Cloud services), the infrastructure provider (which manages the infrastructure) and the Internet Service Provider or ISP (which manages the Wide Area Network infrastructure).

2.2.1 Centralized Clouds

This architecture provides a fixed infrastructure localized in a delimited geographical area to which all clients are connected. According to Bolhuis [114], in a centralized Cloud topology a Cloud provider manages “one or a few datacenters [i.e. infrastructures] located in a small geographical area”. According to Alicherry and Lakshman [115], in a centralized topology, “large datacenters are placed at a few locations”. A centralized Cloud uses an infrastructure managed by the same Cloud provider.

An example of a centralized topology is found in Dropbox [69], which Drago et al. [116] show has a set of datacenters centralized in the United States of America. Slatman [117] claimed the same centralization existed for Microsoft’s SkyDrive in 2013 (precedent of Microsoft OneDrive [81]). Figure 2.3 shows the logical architecture of a centralized Cloud.

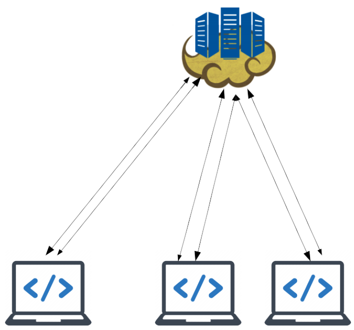


Figure 2.3: Logical centralized Cloud architecture

Centralized Clouds constitute one of the most common commercial offering. According to Babaoglu et al. [118], the centralized model is appropriate for applications such as scientific computations, data mining and Internet-scale Web Services (i.e., geographically distributed).

One of the common issues attributed to centralized Cloud architectures is the, sometimes, excessive distance between the clients and the Virtual Machine (i.e. datacenter). According to Alicherry and Lakshman [115], distributing the datacenters in different locations reduces the latency experienced by users compared to centralized datacenters. Also, according to the same article, distributing datacenters help reducing the network bandwidth needs for high-bandwidth applications.

On the other hand, centralized architectures are also considered to have single points of failure [118,119]. A single point of failure is a component in a system which failure stops the service. In the case of centralized Clouds, a failure in the centralized infrastructure causes the service to be halted.

Because of their size, large datacenters need to be located near a power plant, according to Orgerie et al. [120]. Also, datacenters cannot be dynamically extended when the computation demand becomes too large for their limited resources. An increase in demand may force the computing resources of the Cloud provider to be extended. An extension in the infrastructure resources may imply a modification of the datacenter size and/or design. This is a costly process, as an extension of the datacenter normally requires a physical modification of the building. This implies that datacenters are constrained to their capacity, and cannot dynamically increase it (as increasing their capacity implies setting up new machines).

2.2.2 Distributed Clouds

The issues previously described for centralized datacenters are known to many Cloud providers. For example, Google [121] and Amazon [77] construct their Cloud datacenters in different geographical locations, so they provide a service closer to the end user. This type of Cloud is called a distributed Cloud. A distributed Cloud is formed by several infrastructures in different locations, in opposition to centralized Clouds. Alicherry and Lakshman [115] defined distributed Clouds as Clouds where the resources are geographically distributed in multiple distant locations and interconnected over a Wide Area Network. Garg et al. [122] address it as "multiple datacenters" distributed in different geographical locations worldwide. Finally, Endo et al. [49] define that in distributed Clouds Cloud providers do not own the infrastructure but hire it on demand and use the ISP to obtain dedicated connectivity and resources.

A distributed Cloud is formed by multiple datacenters, i.e. CDN, or by a set of clients which contribute resources to form their own Cloud, i.e. P2P. A review of distributed technologies which compete against centralized architectures is shown in [123]. Another type of distributed Cloud are the social Clouds. Social Clouds are an implementation of a distributed Cloud infrastructure where all the resources used are lent by clients. Over this infrastructure, a platform is deployed. Chard et al. propose in [124], using a social network structure for allowing clients sharing heterogeneous resources in the Cloud to run distributed applications. They also propose adapting a social market place where inherent corrective mechanisms (incentives, disincentives) are used to regulate the system.

Distributed Clouds solve the aforementioned issues of the centralized Clouds. Distributed Clouds are considered to provide a more adaptable and robust system because they do not suffer from single points of failure, as stated by Babaoglu et al. [118]. They also provide a service closer to the client, as stated by Steiner et al. [125]. According to the authors, latency to users can be decreased by distributing the resources in the Cloud. Furthermore, bandwidth can be costs reduced and the availability of the infrastructure increased in distributed Clouds. This is because distance to the datacenter produces a variable access latency due to the long path lengths and going through multiple service providers. The low latency experienced by clients is one of the more praised benefits of distributed architectures [49]. In fact, according to Leighton [1], there is a strong relation between latency and distance. This relation is studied in depth in Section 2.3. Due to the design of distributed architectures, resources in the infrastructure keep a high communication rate between them [50, 51], to propagate changes.

In a distributed Cloud, when a datacenter exceeds its capacity it can dynamically outsource computation to an external datacenter [115]. This way, from a platform point of view, a distributed Cloud can run collaborative applications in parallel using multiple datacenters [126]. For example, Mejiías and Roy proposed the use of a distributed set of small datacenters which is exposed as a single Cloud to the user, which they called mini-Clouds [127]. The authors were able to provide quicker scalability and a more scalable storage than centralized Clouds using their distributed approach.

However, distributed Clouds bring other issues. For example, according to Alicherry and Lakshman [115], the latency in communication between the different datacenters is far more significant than in the case of centralized Cloud architectures. This is because in centralized datacenters the Cloud resources are concentrated inside a few large datacenters. In a centralized Cloud all resources are located nearby, while in a distributed Cloud they are distant. Then, applications with a high volume of internal communication may perform worse in distributed Clouds than in centralized ones, if the instances are placed distantly from each other. These issues have been considered in the existing distributed Cloud approaches studied in Section 2.2.3.

2.2.3 Representative Examples of Distributed Clouds

Distributed Cloud architectures have been widely considered in literature. In this section we evaluate existing distributed approaches from the point of view of the infrastructure used and the platform deployed to provide services. We also identify involved stakeholders in each case.

Content Distribution Networks (CDNs)

According to Sariu et al. [128], Content Delivery Networks use an infrastructure formed by dedicated collections of servers strategically distributed across the wide-area Internet to host and distribute content. According to Bakiras and Loukopoulos [129], Content Delivery Networks create a platform on a distributed infrastructure to deliver data to a set of distant clients. Data delivery is done asynchronously, as users receive data upon request. CDNs greatly reduce the latency experienced by the users by allocating data close to the user. This goal is achieved through the caching or replicating of data across several devices, which later send this information to the clients [128]. When a CDN server receives the data, it redistributes these data to the clients.

Virtual environments can be deployed in a CDN. A user willing to deploy a Cloud platform on a CDN can use the pooled resources of the infrastructure to deploy on demand computation. Infrastructure providers such as Google [95] offer a CDN infrastructure to offer users a service over which they can deploy a platform for distributing content. The schema of operation of a CDN Cloud is shown in Figure 2.4.

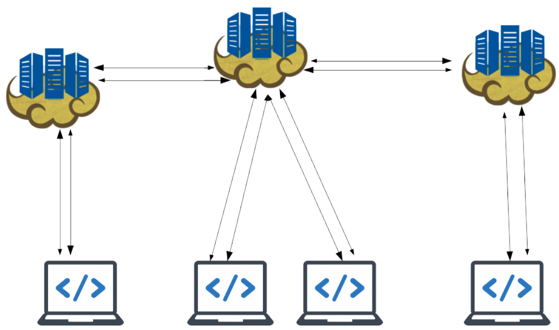


Figure 2.4: Schema of operation of a CDN Cloud

Content Delivery Networks enable content publishers to make their content widely available. The distribution of resources helps provisioning the network access of the servers, their capabilities in terms of CPU and disk bandwidth, etc. Also, resources can be dimensioned according to the popularity of the content in an area and even temporal patterns of access. Unfortunately, content popularity is difficult to predict. Clearly, content publishers cannot provision their sites for sudden load peaks as this is not economically viable, as stated by Passarella [130]. Then, CDN services are most of the time implemented and commercialized by dedicated companies, such as Akamai [131].

According to Passarella et al. [130] Content Distribution Networks are formed by two blocks: Content distribution and replication, and a redirection system. The first block, i.e. Content Distribution and Replication, has two functionalities: identifying the content to replicate and identifying on which servers to replicate each content. The second block, called the redirection system, maps the request for content from the client to its real (replicated) location, i.e. redirect the client's request to the replicated content.

The problem of allocation of popular files has also been tackled in literature. Hu et al. [132] propose a placement algorithm for CDN Clouds. Their algorithm offers a better balance between placement costs and Quality of Service than using a traditional Least Frequently Used algorithm. Their work could be extended by adding more constraints to the placement. For example, the authors do not target restrictions such as energy-efficiency. Also, the authors do not target the Quality of Service from different points of view, such as prioritizing resiliency over latency. For further information, a complete survey on Content Distribution Networks is done by Pierre et al. [133].

P2P Clouds

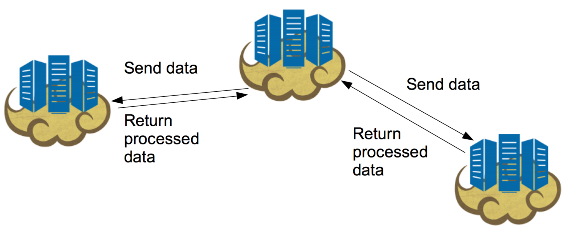
Peer-To-Peer (P2P) Clouds are another representative example of distributed Clouds. According to Babaoglu et al. [118], in P2P Clouds the Cloud provider deploys a Cloud platform using an infrastructure formed out of independent resources. These resources are opportunistically assembled at virtually no cost. Resources can belong to infrastructure providers (datacenters) or users (personal devices). In a P2P Cloud the communication is done between "peers", that is, all clients are treated as equals contributors to the system.

A P2P Cloud operates similarly to a CDN, as it also uses several participant devices to distribute information. The main difference between CDN and P2P Clouds is that P2P Clouds can be used for other tasks than Content Delivery, such as parallel computation [134] or storage [135]. For example, Marozzo et al. [134] study the use of P2P Clouds to solve MapReduce problems. MapReduce is a programming model for parallel data processing used in Cloud computing environments. The authors claim that, through the use of P2P Clouds, they solved the problem of job failures present in centralized architectures, even with high rates of incoming jobs. Due to this, the authors claim that P2P Clouds are more reliable systems for the use of parallel MapReduce computation than centralized ones.

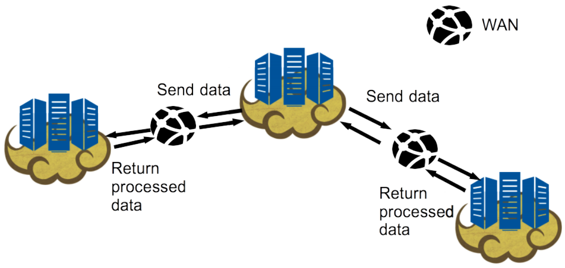
Xu et al. [135] propose the use of P2P Clouds for the storage of data. In their work, they use replication of content through peers to store information in the Cloud. The authors claim that, using P2P Clouds, they resolve the problem of bottlenecks in centralized systems. Another example of P2P Cloud systems for storage is found in the work of Peng et al. [136]. Here, the authors propose a P2P Cloud architecture towards storage and parallel computation. In their work, the authors claim the superiority of distributed networks facing centralized ones. They agree with Xu et al. [135] that a P2P Cloud solves the problems of bottlenecks found in centralized Clouds. Jin and Kwok also addressed the problem of distribution of storage in a P2P network which included mobile devices [137]. In their work, the authors propose a Cloud assisted system architecture for P2P networks (i.e. peers are assisted by a Cloud during content discovery), and targeted energy-efficiency. They use collocation of data, so when a peer is in need of bandwidth and/or low energy it can request others to host the data for it. The authors propose a prediction system for the collocation which, as the authors claim, shows a better accuracy than the means method. The authors demonstrate being able to save energy consumption in the mobile devices to cope with the limited energy capacity of such devices, but they do not consider how much energy is consumed in the Cloud.

P2P Clouds are also useful for content delivery. For example, Li et al. [138] proposed a P2P Cloud for content delivery. In this work, the authors focus on the Optimal Bandwidth allocation Problem in P2P Clouds, i.e. how much bandwidth to allocate in the Peers for the distribution of content. The authors develop an algorithm which they claim maximizes the relationship between allocated bandwidth to specific clients, which act as seeds, and the download rate of all clients.

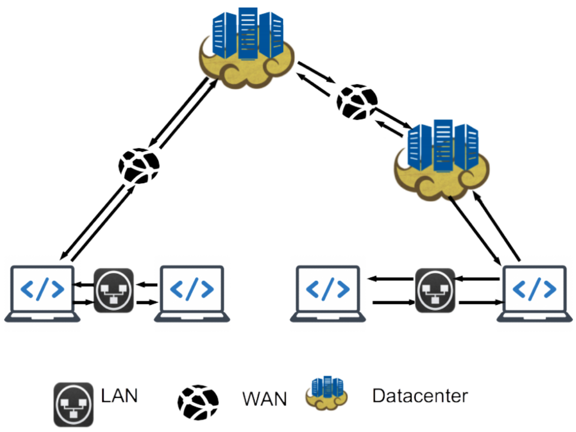
The schema of operation of a P2P Cloud in parallel computing is shown in Figure 2.5a. In a P2P Cloud for parallel computing several Virtual Machines are deployed in different resources. When a Virtual Machine requires parallel computation to complete a job, data is sent to the peer VMs to be processed. When finished, the processed data is returned to the original Virtual Machine. Figure 2.5b depicts the schema of P2P Clouds for storage. In this case, the data are split into different chunk servers. When accessed, data is retrieved from all of them. Finally, the schema of content delivery using P2P Clouds is depicted in Figure 2.5c. In this schema, data are sent to different nodes, which propagate them in the network. When a node requests some content, it accesses the closest peer to retrieve it. After retrieving the content other clients can reach this client to retrieve data.



(a) P2P example schema for parallel computing between datacenters



(b) P2P example schema for Content Storage



(c) P2P example schema for Content Distribution

Figure 2.5: P2P schemas

P2P Clouds also have issues. For example, some P2P architecture may make a heavy use of the clients' resources. This situation leads to users not being motivated to share their own resources, as researched by Hughes et al. [139]. These architectures are also heavy on the networks' broadband, due to excessive replication, as studied by Lv et al. [60] and Rodriguez et al. [140]. Also, because of the existence of different copies of the data, these approaches are prone to consistency issues, unless a high speed communication network exists between the nodes, acording to several studies such as Narayanan et al. [50] and

Coady et al. [51]. In general, distributed approaches are less effective while managing multiple concurrent modifications of the same data from several clients in different locations than other approaches. Also, according to Passarella [130], mobility of peers is also a great challenge for P2P architectures, as P2P Clouds need to establish and maintain an overlay structure of communications. When a peer moves its IP configuration changes, which implies a change in the peer ID. This changes need to be managed in the overlay.

Volunteer Cloud Computing

In volunteer Cloud computing the Cloud Provider deploys a master/worker platform in an infrastructure formed out of multiple distributed devices. A central node (usually managed by an infrastructure provider in a datacenter) outsources computation to users devices (usually personal computers). In volunteer Cloud computing communication is not between equals and the offload of computation is always from the master to the slaves. Chandra and Weissman [141] proposed using distributed voluntary resources to build Clouds. In their work, authors claim that the use of datacenters is unnecessary for some applications. As an example, they claim that Cloud providers offer strong performance guarantees which are not needed for some services, which increases the pricing. According to authors, users may be constrained by the data movement costs to the Cloud. Their approach, called Nebulas, proposes using only resources donated by end-users hosts to manage Cloud services. The authors claim Nebulas to be scalable and offer a low cost of deployment. They also propose a method for it to be more reliable. The authors use variable replication of data to ensure reliability.

Ryden et al. [142] extended the definition of Nebulas proposed by Chandra and Weissman [141] to support data-intensive applications. The authors implementation of Nebulas enables data-intensive computing in the volunteer nodes, exploiting the location information of the nodes to replicate data and place computation. In their work, the authors evaluate the performance of using Nebulas on a simulation of a volunteer network. Their experiments evaluated the performance of MapReduce over the network. The authors claimed that their solution outperformed other distributed volunteer systems. The main difference between their approach and the compared volunteer systems is that, in their approach, input data are distributed, while the compared approaches centralize the input data in a server which distributes data among the volunteer nodes. The authors explain the outperforming of their solution against the compared ones due to the "locality-aware scheduling of computation and placement of data" done by Nebula. The authors also claim that Nebula is "highly robust to both transient and crash failures".

Mobile Cloud Computing

The wide uptake of Cloud architectures caused global IP traffic to increase fivefold in between 2009 and 2014 [33], catalyzed by the ubiquity of mobile devices. According to Cisco, global IP traffic is envisioned to increase threefold over 2014 and 2019, with mobile wireless traffic exceeding wired traffic by 2018 [33]. This adds to the limitations in resources and connectivity of mobile devices [35], bringing new challenges to the Cloud. In order to address the aforementioned limitations, a new paradigm emerged: mobile Cloud computing. Mobile Cloud computing brings new opportunities to support resource-hungry services for mobile devices, like image processing [143], crowd computing [144], Internet data sharing [145], wearable devices' sensing [146] or augmented reality [147], by offloading data and computation into the Cloud [35].

Several definitions have been proposed for Mobile Cloud Computing. Fernando et al. [35] identify the three visions of mobile Cloud computing according to the stakeholders of the infrastructure:

Mobile Clouds running on traditional infrastructure providers. In this definition, a Cloud provider deploys a Cloud on a datacenter, as shown in Figure 2.6a. Mobile users connect to this remote datacenter using wireless connections such as WiFi or 3G. This approach is very similar to the datacenter-centric approach, but the Cloud provider uses multiple datacenters to host the computation closer to the client. However, data has still to travel to "remote servers" [35].

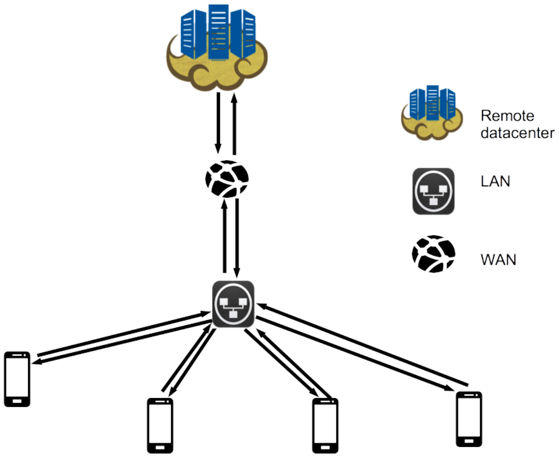
Mobile Clouds running on CloudLets. In this approach, mobile devices offload computation in a series of local domestic servers called CloudLets [145], as shown in Figure 2.6b. These servers have connectivity through a Wide Area Network to a remote datacenter. The difference with the traditional infrastructure providers is that using CloudLets data have to travel a smaller distance.

Mobile Clouds running on available nearby devices. According to Fernando et al. [35], another approach to mobile Cloud computing considers the collective resources of the various mobile devices in the local vicinity, and other stationary devices too if available, as resource providers of the Cloud. This approach considers mobile Cloud computing as a Cloud platform deployed by a Cloud provider in an infrastructure blending multiple devices (mobile devices, personal computers, etc.) and connection types (WiFi, Ethernet, etc.) both mobile and static, as shown in Figure 2.6c. In this definition of mobile Cloud, the Cloud's capabilities are used to offload data and computation off the mobile devices into any available device nearby. This static infrastructure can be either private, such as domestic servers or CloudLets [145], public, interacting with public datacenters, or hybrid. In this approach data does not travel far from the source, which reduces latency [1]. However, when no available datacenter is near the mobile device, the Cloud platform lacks the reliability of such infrastructures.

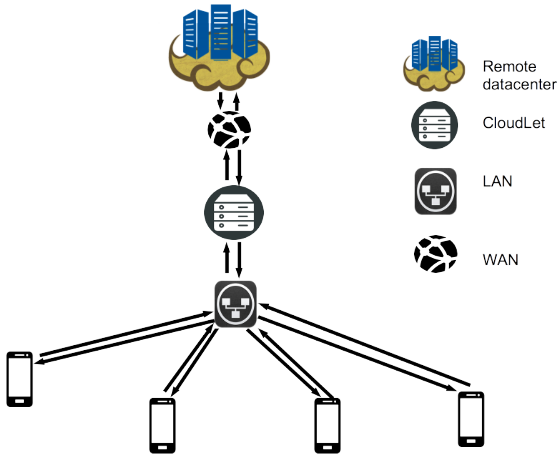
Edge Cloud

Edge Clouds were conceived as a solution to the dependence on large datacenters. Its main purpose is to reduce the distance between the processing of the data and the consumption by the final client. To do so, edge Cloud computing relies on devices on the users' edge of the network. According to Islam and Gregoire [148], an Edge Cloud is "essentially a computing and storage Cloud running a variety of value-added services managed by an ISP in proximity of and for its customers".

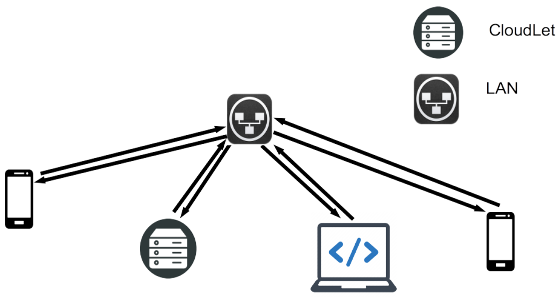
An example of a project which leverages the power of edge Cloud computing is the DISCOVERY initiative [149]. In this project, the authors propose leveraging the computing capacities of network Points-of-Presence available through the Internet. A point of presence is a small or medium datacenter which marks the beginning of the ISP network. They claim that the computation power of these datacenters is under utilized, and it can be used for providing locality-aware Cloud services [26]. However, the authors remark that the design costs of such a solution are high and it should be considered only for deployment over a mature and efficient solutions. As such, the authors propose the development of an operating system, running over OpenStack. Their system, referred to as Locality-Based Utility Computing (LUC) Operating System, is designed for Locality-Based UC [26]. It aims at abstracting the user of such infrastructure, where the locality of the computation plays an important role, and "turning a LUC infrastructure into a collection of abstracted computing facilities that are as convenient to administrate and use as available Infrastructure-as-a-Service (IaaS) managers" [149].



(a) Mobile Cloud platform on traditional infrastructures



(b) Mobile Cloud platform using CloudLets



(c) Mobile Cloud platform running on available nearby devices

Figure 2.6: P2P schemas

Another relevant approach of edge Cloud computing was presented by Valancius et al. [55]. In this work, the authors propose using ISP-controlled home gateways to provide computing and storage Cloud services. Data are replicated in the home gateways, which reduces information traveling on the network. Their solution, called NanoDatacenters, simulates a P2P architecture in the edge of the ISP's network. The authors tested their solution by simulating different populations of users requesting Video on Demand. Through the distribution of data, they claim to save, in the worst case scenario, at least 20% to 30% of the energy compared to traditional datacenters. While their work is highly focused on energy-efficiency, by hosting data in the edge they are equally able to reduce latency. The authors summarize the advantages of their proposal in 4 points compared to centralized datacenters: better heat dissipation, a service closer to the user, a self-scalable architecture and more energy-efficient system. One of the main concerns of the authors is that some users may switch off their home gateways when they are underutilized. However, according to their studies, equipment at the end user is up on average 85% of the time. They acknowledge that, in small populations, the main issue is the limitation on storage in the gateway. This problem is solved when the population is increased. This is because, as the population increases, the data can be sliced in more parts or migrated to gateways with more free storage.

Fog Computing

Fog computing is a recent term, used for the first time by Bonomi [150] in 2011. However, a standard has not being reached yet about its meaning. It was early defined by Bonomi et al. [151] as "a highly virtual platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers". It was first used in the context of connected cars, which need location awareness, a large number of devices and small latency. It was expected by the authors to interplay with Cloud computing.

On the other hand, some authors, such as Vaquero and Rodero-Merino [152] have a different definition. In their work [152], the authors define fog computing as a Cloud where a heterogeneous wireless, ubiquitous and decentralized devices cooperate among them and with the network without the intervention of third-parties. According to the authors, mobile Cloud can be used for providing basic network functions or new services and applications running in a virtual environments. Finally, the authors agree that users leasing part of their devices to host these services should get incentives for doing so.

The main common points across all definitions are the low latency between nodes and location awareness. Also, fog definitions normally address connected objects. We adopt a definition of fog computing that includes the datacenter, the edge and the end-users devices as part of the Cloud. Despite the inclusion of datacenters, Fog computing leans most of the computational weight to the end-user and edge side, as this technology is meant to exploit the computational power of all the devices involved in the communication. Fog computing shares many similarities with mobile Cloud computing, such as the inclusion of end-user nodes in the Cloud. Also, mobile computing and fog computing share many security threats, such as all those affecting the devices which are not managed by large companies (such as mobile phones in the case of mobile Cloud computing) [153].

Figure 2.7 shows the classification of Cloud platforms according to the Cloud Infrastructure. As depicted in the figure, datacenter-centric Clouds run Virtual Machines in distant datacenters. In opposition to it, edge Clouds run Virtual Machines in ISPs managed devices (e.g. ISPs' edge datacenters and ISPs' managed domestic gateways [55]). Mobile Cloud moves the Cloud to the client devices, and utilizes nearby resources. Finally, fog computing tries to integrate all devices (from the datacenter to the client machine).

2.2.4 Benefits and Drawbacks of Different Cloud Platforms

On the one hand, distributed systems can provide a more energy-efficient solution than centralized ones, as shown by Valancius et al. [55] and Jalali et al. [154]. Moreover, they reduce the latency experienced by the user by reducing the distance between the client and the datacenter [35]. Also, in some implementations, distributed Cloud computing can provide a computational power which is either unattainable or high-priced with centralized computing [57]. Finally, distributed Clouds manage better the distribution of content than centralized platforms [130].

On the other hand, some authors such as Chandra et al. [53] argue that the centralized Cloud model is better fitting than distributed Clouds for "largely stateless services with limited data transmission, such as the Web, or for analyzing batch data that originates inside the Cloud, as when mining database transactions.". This is because centralized approaches provide a high computational power within a very small distance. For these applications, internal communication (between servers) is higher than external communication (client to the server). Moreover, every distributed Cloud which involves the collaboration of private devices, such as P2P Clouds, faces the problem of free riding [155]. In such

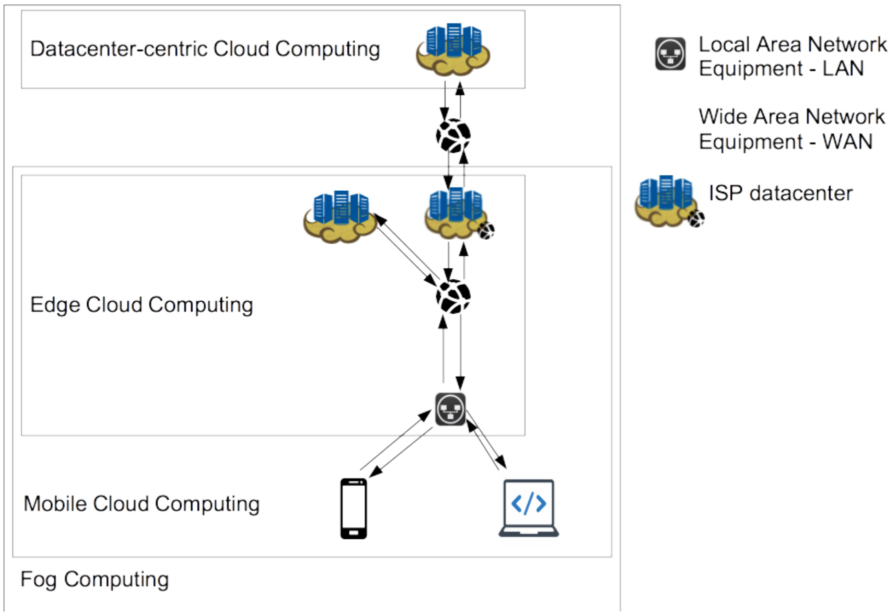


Figure 2.7: Cloud computing technologies classification

networks it is needed the collaboration of multiple parties, which contribute part of their private resources. However, some parties will collaborate more than others, either due to the clients' shortage of resources or a non-collaborative mentality. In consequence, these systems result in unfairness to some users, who contribute more resources than others. Some solutions exist for the problem of free riding in distributed systems in the literature. For example, Zhang et al. [156] propose allowing users to report others which are not fairly collaborating. The problem of free riding has yet to be addressed in the context of very dynamic distributed Clouds, such as mobile Clouds. Research has greatly advanced on the prevention of free riding through applying penalties to users who do not collaborate, but not in encouraging clients to use collaborative systems through the use of incentives.

2.2.5 Comparison of Cloud Approaches

In this section, we compare the Cloud approaches described before in Table 2.2.

We base our evaluation on the following aspects, which have been chosen because of how they affect Quality of Service to the user, and their relation with the characteristics previously defined for Cloud applications.

Target: As defined in Section 2.1.3 we identify different Cloud applications. Each application has different requirements and characteristics. The target of an approach represents which applications it is designed to fit.

Replication: A common tool in multiple systems. For example, replication is used to improve performance [157] and fault tolerance [158]. Moreover, as explained by Abadi et al. [159], replication is a common approach to maintain consistency is to replicate data across several nodes. However, replication can be heavy on the network, especially in networks with limited bandwidth. When an approach replicates content between nodes it is using the available bandwidth. If the replication is too often or the data to replicate are too big, it may require a significant part of the network resources [160]. We divided the approaches into those that use replication and those approaches that manage only one

	Centralized Cloud	**CDN**	**P2P**	**Volunteer Cloud comp**	**MCC**	**Edge Cloud**
Architecture	Centralized	Distributed	Distributed	Distributed	Distributed	Distributed
Target	Web services & data analyzing	Fast Content Delivery	Fast Content Delivery, Storage & Parallel computing	Parallel computing	Offload computation from mobile devices	Content Delivery & closer services
Replication	One copy of data	Yes	Yes	One copy of data	One copy of data	Yes
Adaptability	Limited to the datacenter topology	Limited to the CDN's network topology	Very adaptive	Very adaptive	Very adaptive	Limited to the ISP's resources
Network orientation	No	Limited to the CDN's network topology	Very network orientated	Very network orientated	Very network orientated	Limited to the ISP's network
Network utilization	Low	Medium between the CDN's nodes	High	Medium	Medium in the edge	Low
Locality Mobility	No	No	Yes	Yes	Yes	Limited to the edge infrastructure
Latency	High	Medium	Low	Low	Low	Depending on the ISP's infrastructure

Table 2.2: Comparison of existing architectures

working copy of data (i.e. data are modified always in the same Virtual Machine). We study the relation between latency and consistency in Section 2.3.

Adaptability: While some applications such as web services have static requirements (i.e. do not change requirements of space or computation capabilities often), other applications change dynamically (e.g. real-time applications). We define adaptability as the ability of a system to adapt its topology to changing circumstances such as an increase in the required computational power (e.g. a real-time application may need to extend its resources to keep up with an increase in users without affecting its response time). In some cases the systems have high adaptability. For example, distributed systems where end-users devices are used to host Virtual Machines, such as P2P Clouds, have a high adaptability, as the number of possible host devices increases with the number of clients. On the other hand, in approaches where the infrastructure is deployed by a certain company, such as CDN Clouds, the adaptability of the system is subject to the size and topology of the infrastructure. In the case of edge Clouds, some approaches like NanoDatacenters [55] use an infrastructure composed of a large number of devices (i.e. the ISP owned home-gateways). As shown by the authors, the adaptability of the approach depends on the density of population (i.e. number of gateways). Under normal circumstances this approach is highly adaptable.

Network awareness: Some Cloud applications make heavier use of the ISP network than others. In this characteristic we evaluate the orientation that the Cloud platform has towards the network infrastructure interconnecting the client and the Virtual Machine. Similar to adaptability characteristics, the network orientation depends on the network topology managed by the Cloud platform. Centralized approaches are not network oriented, as they only manage the network inside the datacenter. On the other hand, P2P networks are network oriented, especially if they are formed by volunteer end-user devices, as they take into account the network infrastructure between clients to construct the Cloud platform topology (for example, while finding a content provider). In the case of an edge Cloud, the approaches are network oriented inside the ISP network.

Network utilization: The ISP network requirements of different applications vary. Some Cloud platforms make a larger utilization of network resources than others (for example, for management or replication purposes). Applications which make heavy use of the network may see its usable bandwidth reduced by the system's network utilization. Thus, the Quality of Service of the user may be affected. This characteristic evaluates how heavy is the use that the system makes of the network. This is more relevant on network-limited technologies, such as mesh wireless networks [161] (i.e. radio-based networks which deliver data among multiple clients, normally in areas where wired access is complicated). In this case, the utilization of the network's broadband needs to be as small as possible. Furthermore, if the system runs on a static network it can be assumed that this network is shared with other services, thus making it also important to reduce the broadband utilization. We classified centralized approaches as low utilization (i.e. they do not use the external network for management). On the other hand, we classified as high these approaches which are heavy in the network. For example, some P2P approaches may make a heavy use of the network for replication.

Locality mobility: Some applications need to be aware of their location. In the case of distributed systems, the location of a virtual environment is not only determined by the resource hosting it, but also the geographical location of the host. In a Cloud platform location may change (e.g. in a distributed Cloud a Virtual Machine can be migrated between hosts). The capacity of a virtual environment to be located in different geographical locations is evaluated here.

Latency in communications: This characteristic evaluates how fitting is the system for latency-critical applications. In some applications such as document editing applications, users request low latencies to avoid having users working under different versions of the document, as explained in Section 2.1.3. As shown in Table 2.2, mobile Cloud computing shares multiple characteristics with P2P, but a different target. Also, mobile Clouds do not rely on replication of data, but keep one copy of data active at every moment. On the other hand, differences between edge computing and mobile Cloud computing are depicted. These differences are created by the difference in infrastructures, as edge Clouds are confined to edge infrastructures. For example, edge Clouds are less adaptive than mobile Clouds when the latter uses an approach where it exploits available nearby devices.

2.2.6 Discussion

Edge and mobile Cloud computing are technologies which have raised a lot of research interest in the last years. As shown in the previous section, these architectures reduce the latency between the user and the service (provide a service closer to the end-user). They also reduce amount of data in the ISP network and reduce the energy consumption [55]. Roman et al. [153] provide a complete survey comparing Edge Computing and mobile Cloud computing. It is shown in their work that mobile Cloud Computing is similar to Edge Clouds. Both approaches aim at bringing the Cloud closer to the final user, exploiting unused computation of nearby devices. Notwithstanding, they also exhibit the following differences.

First, mobile Cloud computing assumes a total independence of the Cloud platform from the location of the mobile device, supporting user mobility, in its approach running on nearby devices [35]. Whenever a mobile device needs to be offloaded of computation it communicates with a close by VM. This is not the case in edge Cloud, where clients are attached to an infrastructure. According to Roman et al. [153], computation is run either in the ISPs' infrastructure or a close by datacenter. Thus, edge Cloud computing centralizes information in a static infrastructure, while mobile Cloud computing provides a mobile infrastructure which moves with the client.

Second, from a security point of view, both solutions share several points in common. According to Roman et al. [153], the problem with enabling technologies, such as wireless networks, distributed and peer-to-peer systems and virtualization platforms, is a shared concern in all edge solutions. For this problem, the authors propose the creation of a unified and transversal view of the security solutions of the different technologies. However, the authors note that mobile Cloud computing shows a set of security issues of its own, in addition to these of edge Clouds. First, the authors show that in these architectures were user-owned devices contribute resources, i.e. mobile Cloud computing, a scenario is created where the malicious adversaries can deploy their own gateways (i.e. a Man-in-the-Middle attack) [153]. Also, the authors notice that security threats increase in these solutions where the infrastructure is not managed by large companies, i.e. mobile Cloud computing, due to issues like deficient maintenance or lack of physical protection [153].

Third, an edge Cloud is designed to provide Cloud services from an established infrastructure (for example, the ISPs' datacenter or other infrastructure) [153]. However, a mobile Cloud computing can be designed as a platform relying on the local available resources. Indeed, according to Fernando et al. [35], future mobile Clouds will be a hybrid approach where the infrastructure is lent by users, which act as Cloud resources but at the same time ensures the ability to connect to remote servers, which would provide a higher computational power than the users' infrastructure. This connection to remote servers would be restricted, however, to these cases where the user has a good connectivity and other conditions such as access fees, available battery, and response time with the datacenter. Then, while edge Clouds are always deployed in the form of a nearby infrastructure, mobile Cloud computing can use nearby peers.

Finally, edge Cloud may face problems with the interoperability between Clouds. An edge Cloud is implemented by a specific provider (i.e. ISP), according to Chang et al. [162]. This definition of edge Cloud is consistent with the definition given by Roman et al. [153]. An edge Cloud is managed by a Cloud provider using a providers infrastructure. If a VM needs to be migrated to a different Cloud, i.e. a different provider's infrastructure, it requires both providers to collaborate. On the other hand, since mobile Cloud computing is designed to work on heterogeneous machines but managed as a single platform [35], the interoperability of Clouds is easier.

We have stated before that latency is an important variable for several types of applications. Also, we have stated that distributed Clouds offer a lower latency than centralized Clouds. In order to have a better understanding on the relation between Clouds in the edge of the network and Quality of Service, we study how the network affects the Quality of Service in the next section.

2.3 Quality of Service Requirements

We understand by Quality of Service (QoS) of an infrastructure a set of variables that describe the satisfaction of the user receiving a service in an infrastructure. These variables include latency or transmission delay, availability of the service, error rates in the network and bandwidth. Quality of Service may be evaluated differently by different users, as they may not share the same needs [163]. In this section we focus on how the network infrastructure affects the latency.

2.3.1 Latency Critical Applications

Latency - the time interval between the moment when the client sends a request until it receives a response from the servers - plays an important role in Quality of Service [36]. In fact, some authors give latency more importance over other variables. For example, Delaney et al. [164] address latency as the "single greatest contributing factor to spatial and temporal inconsistencies experienced by end users in the virtual world ". Indeed, a 2009 study for Siemens [36] demonstrated that latency has an important economic effect for small and medium sized businesses.

For these Cloud applications that need a fast response, such as real-time applications (e.g. document editing applications), the latency becomes crucial [94]. Latency may produce consistency issues in real-time applications [164]. These errors may come from two or more clients experiencing different latencies. For example, let's assume two users are working on the same file. If the first user experiences low latency, then all the updates that this user sends to the application are processed fast. On the other hand, the second user is experiencing a higher delay, and receives the modifications of the first user several seconds after they were sent. This case may cause inconsistency issues since, if the changes have not been propagated to all clients, both users are working on different versions of the file. While this situation could be solved by forcing the same latency on both users (i.e. increasing the latency in the first user), it has been shown that increasing the latency as few as 100ms can make users reject the application [165]. Furthermore, Abadi [159] claims that there exist a trade-off between consistency and latency, i.e. it is not possible to have high consistency and low latency at the same time.

Nowadays, centralized Cloud architectures store data in one or more datacenters, which are distant from the client. This architecture is designed to keep a central point to where clients from different locations can share information. However, users experience a detriment in the Quality of Service, as the process of communication adds extra latency to the system, compared to a desktop-based application. Quality of Service is affected, among others, by two factors: bottlenecks in the network and distance between the client and the Virtual Machine.

2.3.2 Distance impact on Latency

The centralization of Virtual Machines in datacenters offers a location where several users share data. For example, two users working on the same instance of an application may use a datacenter which is equidistant from both. However, in many applications, data are not propagated far from where it has been created most of the times. This situation has been studied in literature. For example, Scellato et al. [166] found that 40% of steps in social cascades involve users that are, on average, less than 1,000 km away from each others. A social cascade is a process where data is propagated when a user shares information from another one (e.g. re-tweeting in the microblogging social network Twitter). In a similar manner, Chat et al. [41] study the propagation of data in the Flickr social network. They determined that data can be reached within a few hops (on-line connections between users). Finally, Hristova et al. [40] also investigated the relations between social relations in the Internet (Facebook) and proximity. The authors found a correlation between distance and data sharing (users which live close by share more interests).

The relation between distance and Quality of Service has been studied in literature. For example, Leighton [1] evaluated the relation between distance and Quality of Service by downloading a video in different networks. The author's evaluation is based on four

Distance from Server to User	Network Latency	Typical Packet Loss	Throughput (quality)	4GB DVD Download Time
Local: <100 mi.	1.6 ms	0.6%	44 Mbs (HDTV)	12 min.
Regional: 500–1,000 mi.	16 ms	0.7%	4 Mbs (not quite DVD)	2.2 hrs.
Cross-continent: ~3,000 mi.	48 ms	1.0%	1 Mbs (not quite TV)	8.2 hrs.
Multi-continent: ~6,000 mi.	96 ms	1.4%	0.4 Mbs (poor)	20 hrs.

Figure 2.8: Relation between distance and QoS (Source: [1])

variables: Network latency, packet loss, quality of image and download time. Figure 2.8 shows the results of the author's experiments.

As expected, the longer the distance to be traveled by the information, the more degraded was the information. Leighton [1] explains these results by two main factors. First, latency and throughput (i.e. transmission rate) are tightly related, at a network protocol level, i.e. the protocol used influences latency and throughput. For example, TCP allows only the transmission of a certain amount of data (burst) each time. This means that the device hosting the Virtual Machine waits until the client has received some data before sending the rest. As a consequence, when the distance increases, the delay between communications increases and the throughput decreases. Second, packet loss also affects the latency in transmissions. The device hosting the Virtual Machine waits for the client to send a confirmation of reception of all packets before sending the next burst. If some packets are lost, then they have to be resent. Since the number of packets lost increases with distance, the time between bursts is equally affected.

2.3.3 Bottlenecks Impact on Latency

Latency is not the only QoS variable which is affected by the distance between the client and server. During a client-server communication in a Wide Area Network, data are sent across several nodes. When the distance to be traveled is long, data normally crosses different and heterogeneous networks. If the data travels through different countries, data may be sent along different ISPs' networks.

The Quality of Service along a path which crosses different networks may be affected by different situations. For example, networks may be congested, which adds delays to transmission [167]. Also, as networks are managed by different entities, some networks in the path may have a more ancient infrastructure, which reduces the bit rate [168]. Finally, data may be redirected to a further Internet exchange point, where it can travel through a different network, due to the infrastructure's design. This process is called Internet transit, and is carried out between two or more ISPs through peering [169]. In ISP's exchanges, data can be exchanged through Internet eXchange Points (IXPs) or Commercial Internet eXchange (CIX). Those points are implemented as physical infrastructures where different networks exchange data through external networks [170]. Thus, similarly to distance, bottlenecks play a fundamental role in providing good QoS.

As expected, state-of-the-art equipment in all the routes (and networks) is necessary to obtain a good QoS in terms of latency, errors and bit rates. As stated by the International Telecommunication Union (ICU), the United Nation's agency in charge of information and communication technologies, "high fault report rates [i.e. low Quality of Service] generally indicate a need for improved network equipment" [171]. In fact, network modernization greatly affects quality of service. Regarding modernization of networking equipment, the

ICU uses the following example [168]: “In the UK by September 1989 the main incumbent operator could repair 86% of faults in 1 working day, compared with 65% in September 1988. The improvement was attributed to network modernization spurred by competition. At about the same time the operator, under pressure from the regulator, started to pay compensation for excessive times for service supply and fault repair. If the committed time for service supply or fault repair was missed by 2 working days, customers were paid GBP5 (USD8.4) per working day, with up to GBP1000 (USD1700) for residential customers and GBP5000 (USD8400) for business customers if they suffered financially.”.

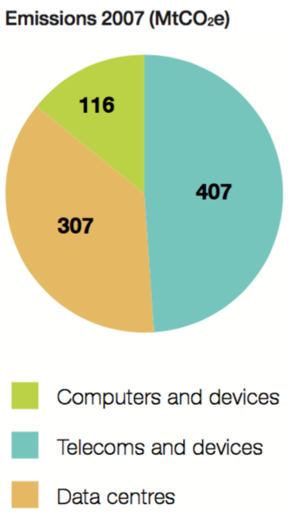
However, this is not always the case in real-life scenarios. In these cases where one or several nodes in the route are slower than average (i.e. old equipment) the overall Quality of Service to the client may be affected [168]. Furthermore, the problem of saturation of bandwidth is also to be considered. As explained in Section 2.2.4, data replication may result in heavy demand on the network, which causes bandwidth issues. When one or more applications make an irresponsible utilization of the network, flooding it with data, the rest of applications receive a smaller bandwidth. Having a smaller bandwidth affects the bit rate of the applications and, consequently, their latency. Bandyopadhyay et al. simulated the effects of network flooding in Mobile Ad-Hoc NETworks (MANET) and showed that it significantly affected packet loss, overhead in communications and bandwidth requirements [172]. To solve this, Sun et al. [173] propose aggregating packets in MANET networks to reduce delay. However, updating the network capabilities does not necessarily avoids the bandwidth issues, as users can also saturate the network. Zheleva et al. [174] showed that, as a network is upgraded, users catch up with the new performance and start requesting heavier services. They showed that, when network capabilities were upgraded in a village in rural Zambia, network performance deteriorated dramatically in time: the average round-trip doubled, the number of bytes associated with failed uploads increased by 222% and failed downloads by 91% because the traffic increased. Independently from a network’s hardware, the amount of data going through it is equally important.

Summarizing, QoS is affected by distance between the client and server, but also by the bottlenecks. To address the problem of bottlenecks and distance in the route, several routing protocols exist, such as Border Way Protocol (BGP) [175] or Open Short Path First (OSPF) [176]. Both protocols are based on a dynamic analysis of the connections between nodes, in order to choose the local best in terms of bit rate. Also, it is possible to introduce static information, in the form of weighted connections, to provide external knowledge to the decision making process and balance data [177].

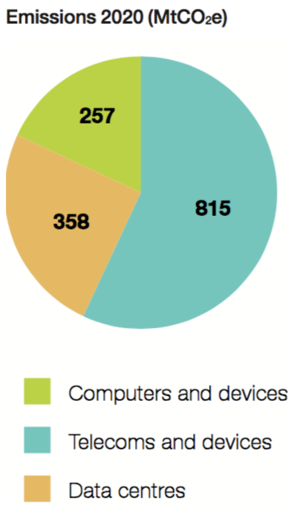
A good Quality of Service is a main factor for the acceptance of an application from a user [165]. On the other hand, in recent years energy consumption of Internet technologies has raised concern [178]. Some authors, such as Gelenbe and Lent [179], consider that a trade-off exists between Quality of Service and energy consumption. In Section 2.4 we evaluate energy consumption in the Cloud. We also evaluate existing solutions to reduce energy consumption.

2.4 Energy Consumption in Cloud Computing

According to GreenPeace [178], the ICT sector is responsible for more than 2% of greenhouse gas emissions. As depicted in Figure 2.9a, together networks and datacenters account for more than 3/4 of the total carbon emissions of IT. In Figure 2.9b is shown GreenPeace’s expected carbon emissions of IT in 2020. As shown, CO_2 emissions of telecommunication networks is expected to double by 2020. Furthermore, Figure 2.9 shows that telecommunication networks are responsible for a great part of carbon emissions. The greater emissions on telecommunication networks are explained because the network resources are kept fully functional for long periods. In many cases, ISPs are forced to keep them running, as data are traveling towards other datacenters and cannot downgrade their resources. However, there also exists another important reason for this situation: telecommunication networks have been obliterated in energy consumption research [180].



(a) GHG emissions of ICT sector as of 2007



(b) Expected GHG emissions of ICT sector in 2020

Figure 2.9: GHG emissions of ICT sector in Million Tonnes Carbon Dioxide (Source: GreenPeace)

In Section 2.4.1 we introduce the motivation for energy efficiency in Clouds. Section 2.4.2 discusses where energy is consumed in the Cloud and Section 2.4.3 presents the most relevant works in energy consumption.

2.4.1 Motivation

In 2013, datacenters consumed 1,500 Terawatt hours of energy, representing 10% of global power consumption, according to Digital Power Group [29]. This represents 50% more than the global aviation, until then to be the most energy consuming ecosystem. Furthermore, most of this energy is used for other purposes than computation. According to the ABB Group [181] an average datacenter of 465 square meters uses 30% of its energy consumption only for cooling, which translates into $281 million a year. Although some of this energy consumption is necessary for the correct performance of the Cloud, most of the energy is wasted because of inefficient utilization of energy, such as idle servers (which are ON but not running computation) or inefficient cooling systems. To measure the energy efficiency of a datacenter is used a value called *Power Usage Effectiveness (PUE)*. PUE measures the energy used on anything that isn't considered a computing device in a datacenter, such as lighting, cooling, etc. An ideal PUE is 1.0, i.e. all energy consumed by the datacenter is used in providing computing services. The PUE of a datacenter is obtained by dividing the *energy consumed in the datacenter* by the *energy consumed by the computing devices*.

The large energy demand caused by new technologies does not only affects the environment, which receives the added carbon emissions caused by the generation of the energy. It also affects the companies' budget, which needs to pay for this energy, and the user, who is affected by the decisions of the company (for example, when the company decides to reduce the equipment to save energy, the user perceives a decrease in the QoS). According to ABB, companies could save $281 million in cooling unused resources [181]. In fact, according to a study conducted by the Natural Resources Defense Council, most datacenters operators are satisfied if achieving a Power Utilization Effectiveness of 1.65 [182]. This means that all these providers are willing to spend an extra 65% of the energy used in the devices in cooling and distribution of energy.

The energy efficiency of datacenters has increased since the emergence of Cloud computing [183], due to its better use of resources. However, recent energy consumption studies [25] show that this trend discontinued - and even reversed in some cases - in the last years. This is explained because some energy efficiency strategies (studied in Section 2.5) are not correctly adapted by the infrastructure providers, but also by an increase in the users demands. Having more users connecting to the Cloud from different locations on the world, different devices and different timezones, complicates the adoption of some of the energy efficiency advances that datacenters can implement, as discussed in [183]. For instance, datacenters need to provide a round clock service to host incoming requests from distant users, that is, datacenters cannot switch off these many servers while they are not utilized, as they may be requested. Furthermore, while networking is basic for the correct functioning of the Cloud, the relation between Cloud Computing and the involved network infrastructures - all the networking devices which are used to communicate with and within the Cloud - have been neglected in literature, as it is shown in works from Buysse et al. [184] and Orgerie et al. [120].

2.4.2 Improving Energy Efficiency in Clouds

In the last years, several research works have been focused in making efficient use of energy in Cloud computing environments. For example, Beloglazov et al. [185] propose a set of VM placing algorithms based on energy-efficiency heuristics to reduce energy consumption in datacenters, while complying with the Quality of Service. In their work, the authors decide on admitting a request for a VM regarding the current VM allocation in the infrastructure, i.e. they only admit new VMs in the Cloud if the new VM can be placed efficiently. Through this, the authors claim an improvement of more than 10% of the energy used in a placement algorithm with no Power-Aware policies.

Also, Baliga et al. [186] considered the energy consumption problem as a compound of datacenter and end-user consumption. To do this, the authors added the energy used for transporting data to and from the datacenter, i.e. the network consumption. They evaluated three different service models: storage as a service, i.e. storage applications, software as a service, i.e. data editing applications, and processing as a service, i.e. scientific computation. Baliga et al. offer a detailed model of energy consumption in the Cloud, including network and end user consumptions for the three different services. They ran experiments on the three services, comparing the energy consumption with the number of users per server, and the transmission rate. They conclude that, in some cases, the energy consumption of using the Cloud becomes higher than using applications running on the end users' devices. This work was the first in considering the Cloud as a whole, including the network and end user. However, the authors did not take into account energy consumption improvements, such as switching OFF unu sed resources, which may affect the allocation of the VM. Moreover, they used a datacenter-centric vision, but did not compare with other architectures, such as distributed or Edge Clouds.

To improve energy efficiency in Clouds it is important to understand where the energy is consumed in the Cloud. In Section 2.4.3 the different elements where energy is consumed in a datacenter-based Cloud is shown.

2.4.3 Cloud Energy Consumption Model

Figure 2.10 shows a schema of the connections between a datacenter and the users and between datacenters of the same Cloud provider in a typical Cloud structure. The infrastructure used to interconnect the user and the datacenter will be addressed as the intermediate infrastructure. It includes all the networking devices used to interconnect datacenters and users, such as ISPs' routers, switches, repeaters and amplifiers, but also networking devices belonging to the Cloud provider, used to interconnect datacenters, such as routers and switches. The connections are, typically, wired, using different technologies such as copper-wire Ethernet or optical fiber.

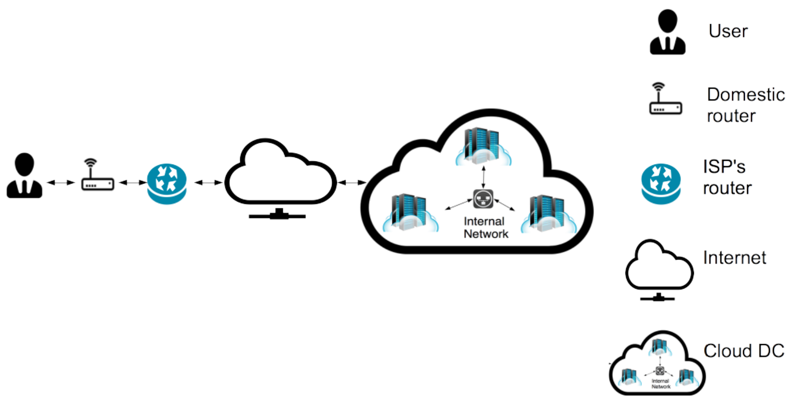


Figure 2.10: Scheme of typical external Cloud interconnection

Figure 2.11 shows a schema of the connections between servers inside a datacenter in a typical datacenter structure. In this schema, referred as the internal infrastructure, are included all the servers and storage devices, but also internal networking devices such as top-of-the-rack switches and hubs; and internal routers. As before, the typical connections are wired and use technologies such as copper-wire Ethernet or optical fiber.

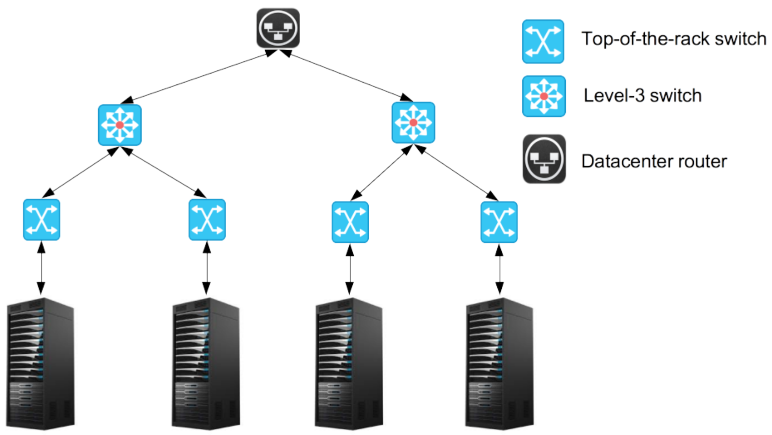


Figure 2.11: Scheme of typical internal Cloud interconnection

Finally, Figure 2.12 shows a schema of the classic connections at the user's side. This schema is referred as user's infrastructure, and includes user's private devices, such as laptops, smartphones and connected objects, and domestic routers. The most used technology in this schema are wireless connections, such as WiFi or 3G/4G. However, it may also include wired connections.

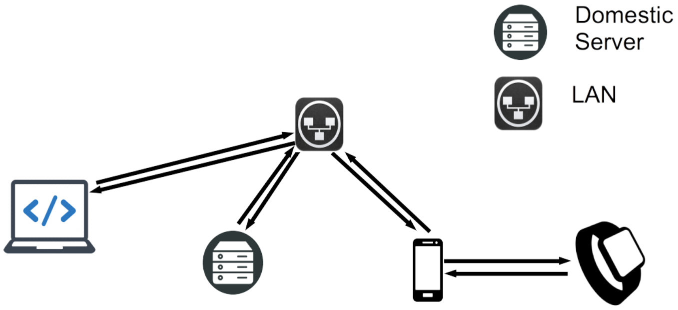


Figure 2.12: Scheme of typical domestic connection

In the following sections, we analyze energy consumption in the Cloud from the three points of view defined before: datacenter infrastructure, intermediate infrastructure and user side. Figure 2.13 shows the datacenter - client architecture and its distribution in energy consumption.

Datacenter Infrastructure

The amount of energy consumed inside a datacenter has been widely modeled in literature. Krioukov et al. [2] analyzed energy consumption in a datacenter. An evaluation of the distribution of the energy consumption in datacenters made by the authors of the paper is shown in Figure 2.14.

As depicted in Figure 2.14, servers are the element which consumes most energy inside a datacenter. However, network equipment accounts for 30% of the total consumption themselves. Infrastructural energy consumption (associated to illumination, cooling, etc.) has not being included in this analysis to ease comparison between servers and networking devices (such as access, aggregation and core switches).

Other useful models can be found in bibliography. Heath et al. [187] propose a linear consumption model for servers, centering on the main memory, data storage, and networking components. Also, Sun et al. [188] study energy-aware resource management, in relation with thermal dissipation, and how it affects performance in heterogeneous datacenters. The authors show that the problem of statically placing servers in a datacenter is NP-complete, and present a scheduling framework to correctly resource management. For further information in energy consumption in datacenters, a survey on servers energy consumption is provided by Mobius et al. [189].

On the other hand, energy consumption of networks inside datacenters has also been studied. Orgerie et al. [190] determine that the energy used by a network device (for example, a router) connected to a network depends on the bandwidth used by the transfer, the transfer duration and the type of equipment, independently if it is being used to transmit data.

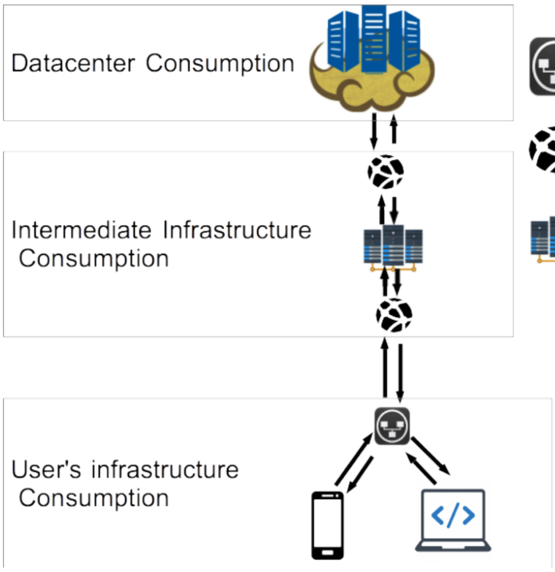


Figure 2.13: Datacenter - client consumption

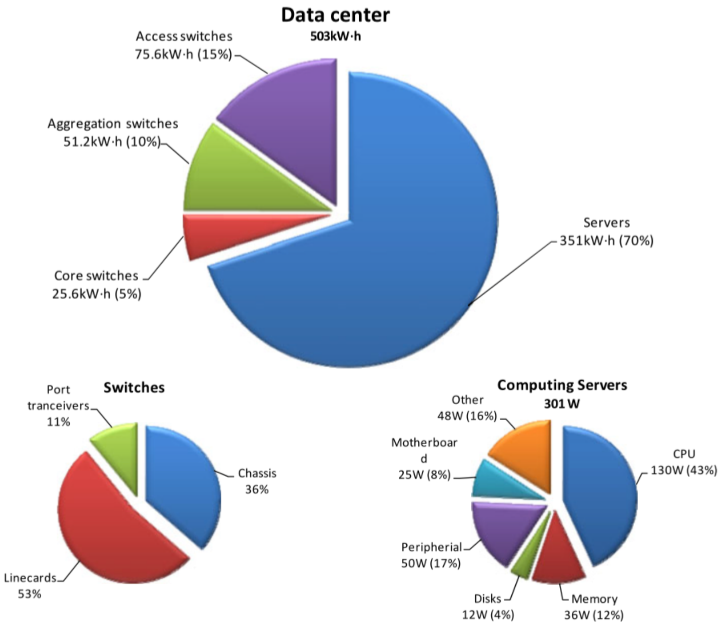


Figure 2.14: Distribution of energy consumption in a datacenter, according to Krioukov et al. [2]

Intermediate Infrastructure

The energy consumed by a network has been less studied in literature than the energy consumption in datacenters. For example, Lange et al. [191] evaluate the energy consumption of telecommunication networks, both from the ISPs' side and inside domestic networks. The authors provide a study on networks' consumption based on traffic and energy consumption data from literature. Then, the authors compare between the use of energy made by both sides (ISP network and domestic networks) and propose different metrics for the evaluation of energy efficiency in networks and opportunities in the field of energy efficiency in telecommunication networks. In particular, the authors give more importance to a load-adaptive approach, i.e. dynamically keeping active only these resources necessary to manage the current load of traffic. However, the modeling of the energy consumption is marginal to their work, and the authors do not provide a formal model.

On the other hand, Gomez et al. [192] provide a formal and complete model of the energy consumption patterns of WiFi and WiMAX gateways. However, in this work, the authors provide a detailed model for specific technologies. The specificity of the model makes it unsuitable for wide area networks such as the network interconnecting the user and the datacenter in the previous schema. A complete survey about energy consumption in networks is shown by Cervero et al. [193]. In this work, the authors evaluate static and dynamic schemes, such as network planning and traffic-engineering, to reduce the energy consumption of networks. They also highlight the main research challenges in the implementation of such schemes.

User's Infrastructure

The energy consumed by a single client can be considered negligible in the overall system (compared to datacenters and networks). For example, according to Apple's specifications, MacBook Pro laptops between 2008 and 2013 consume between 200 and 300 W [194]. In fact, more modern laptops have consumptions of less than 100 Watts (W) [195]. In comparison, an average 5000 feet datacenter consumed 1,127 kW in 2009, according to Emerson Network Power [196], i.e. around 1000 times more.

However, a Cloud application is normally not used by a single user, but a large number of them. Given the amount of users that make use of a Cloud application at the same time, the consumption in the user's side adds up to a considerable amount. In fact, energy consumption of end-user ICT in residential areas has been described as the largest consumer of electricity in the ICT sector, as pictured in Figure 2.15. On the other hand, most of this consumption is spent on entertainment devices. In fact, televisions alone account for more than 50% of this consumption [3]. Within the commercial ICT, end-user consumption of computing devices (laptops, computers and mobile devices) roughly consume the same as datacenters. As shown in Figure 2.15.

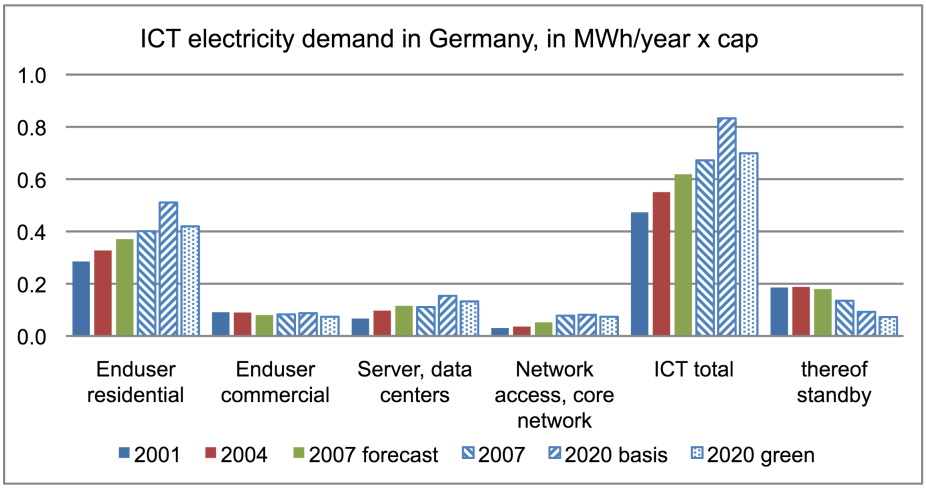
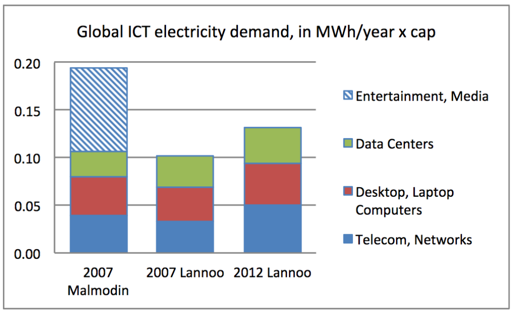
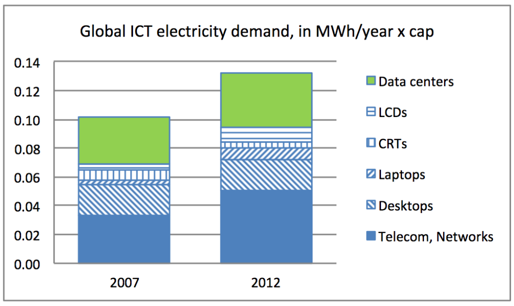


Figure 2.15: ICT electricity demand per capita in Germany 2001-2020, estimated in 2009. Source: [3]

In detail, energy consumption of end-users devices roughly accounts for a third of the overall consumption, as shown in 2.16.



(a) Worldwide ICT electricity consumption per capita, in MWh/cap per year. Source: [3]



(b) Worldwide ICT electricity consumption per capita in still greater detail, in MWh/cap per year. Source: [3]

Figure 2.16: Worldwide ICT electricity consumption per capita. Source: [3]

Figure 2.16a represents the global electricity consumption in ICT per year both in 2007 and 2012, in datacenter, network and end-user (laptop and desktop computers) sides. It shows a similar consumption on the three sectors, once taken away the energy consumption of entertainment devices. Figure 2.16b shows a more detailed graph on the same years. It divides the end-user consumption in screen (LCD/CRT) and computer (laptop/desktop). As it is shown, most of the energy demanded in the end-user side is consumed by desktop computers in year 2012.

Existing models describing energy consumption in the end-users side consider the energy consumption of the household as a whole (including computers, appliances, etc) [197]. This is because ICT end-users models consider appliances such as entertainment devices as part of the end-user consumption, as shown in Figure 2.16. To the best of our knowledge, the amount of energy consumed by a Cloud application on the client's side has not being isolated in the literature.

2.5 Energy Efficiency Improvements

Many works have focused on improving energy efficiency of Internet-based technologies. In this section, we evaluate these improvements which affect the energy efficiency of Cloud computing. We have divided these methods according to the infrastructure they address: Cloud datacenter or ISP infrastructure. Section 2.5.1 discusses the main solutions for energy efficiency in the datacenter. Section 2.5.2 shows the most relevant energy efficiency improvements in the network.

2.5.1 Energy Efficiency Improvements for Cloud Datacenters

Energy efficiency improvements in datacenters have been deeply studied in the literature. According to Orgerie et al. [198], three different types of energy efficiency improvements in the datacenter can be found:

- Infrastructure improvements: It includes improvements implemented on the datacenter environment, such as better refrigeration or distribution of devices.

- Hardware improvements: It includes improvements implemented on the devices' hardware, such as the use of low power components.
- Software improvements: It includes improvements implemented on the devices' software, such as start-stop algorithms.

Infrastructure Improvements

The most relevant solutions for energy efficiency in datacenters are:

Correct placement and isolation of the devices: Correctly placing all the servers in the datacenter is important to save energy. The cold hall, hot hall architecture [199] distributes racks in a line so servers use the cold air in a hall (cold hall) and throw it to another one (hot hall). This way, all servers refrigerate through air in a cold hall and expel the hot air in a hot hall. It is also important to correctly isolate the halls so that the cold air is not lost and the hot one does not interfere. This solution helps reducing the amount of cool air needed to regulate the servers. Other solutions include planning the flow of cold air along the ceiling, solving the extraction problem or placing the devices with a larger load closer to the cold stream [200].

Use of local and green sources of energy: A large amount of energy is lost during its transportation. Datacenters locate themselves close to the electricity central, to avoid loss. Also, using protocols like "follow the sun, follow the wind" [201], datacenters can make use of renewable energies avoiding the problem with changing environments. For instance, a datacenter can be located in a sunny place and, when it is necessary, such as during the nights, the data can be migrated to a datacenter impulsed by wind electricity resources. Van Heddeghem et al. [202] investigate the use of green sources in distributed datacenters. The authors determine that, by locating datacenters near renewable energy sites, and migrating jobs and data it is possible to reduce the footprint of global datacenters by 14%. The authors remark that, in these datacenters where the load is below the peak capacity, i.e. the resources are most of the time underutilized, savings of up to 60% to 90% are possible by applying the "follow the sun, follow the wind" protocol.

Free cooling sources: Many companies solve the cooling problem by placing their datacenters in naturally cold locations to use free cooling. For example, Google placed some of its datacenters in Finland, to leverage the cold weather [203]. In case it cannot be placed in a cold location, because of accessibility or communication reasons, it is also possible to use a datacenter near the seaside and use seawater as a natural cooling resource [120].

Hardware Mechanisms for Improving Energy Efficiency

Hardware manufacturers lately increased their efforts and budget dedicated to provide energy efficient devices. The US program's standard Energy Star works to certify energy efficiency of hardware. According to this organization, adopting new hardware improvements can show differences of up to 54% of energy consumption between two servers [204]. According to Energy Star's study, "common slot power supplies that are redundant, better DC voltage regulators, [..] processors that consume less power, and less power needed to operate cooling fans" are some of the improvements achieved by newer servers facing old ones.

Other improvements present in new hardware systems are ON/OFF and reduction of speeds in servers' components:

- **Hard-disk:** ON/OFF and variation of rotation speed systems [205].

- **Network Card Interface (NIC):** ON/OFF and variation of transmission rate and protocol systems [62, 206].
- **CPU:** ON/OFF and variation of performance frequency systems [207].
- **Motherboard:** ON/OFF systems [120].
- **RAM:** ON/OFF of memory banks systems [208].
- **Fan:** ON/OFF and variation of rotation speed systems [209].

Saving energy by shutting down unused devices is a widely used solution studied in the literature. For example, Lefèvre and Orgerie [210] present their Green Open Cloud (GOC) architecture, an energy-efficient framework for the Cloud. This architecture achieves energy efficiency by switching off unused resources. In order to optimize this process, they also predict resources usage, to switch on resources which will be used in the near future, and aggregate some reservations, to avoid frequent on/off cycles. In their work, the authors show by simulation that by using their framework they are able to save 25% of the Cloud nodes' electrical consumption compared to basic Cloud resources management systems.

Gallego et al. [211] also targeted switching off unused resources. In their work, the authors propose combining virtualization, Dynamic Voltage Scaling (DVS) and switching off unused resources to reduce energy consumption. Through virtualization, the applications can be hosted in the most energy efficient devices. Then, data can routed from and to these devices. This way, unused resources can be switched off. They use a proactive-reactive method which takes decisions dynamically using the best practices of both reactive and proactive mechanisms. The authors evaluated their solution through simulation of synthetic data. By using their method, the authors claim to have a significant energy saving compared to pure proactive and pure reactive methods, while meeting the same Quality of Service.

As an extension to ON/OFF and reduction of speed, a relevant hardware improvement is *Sleep mode.* In this mode, components consume less than when they are ON, but are ready to be switched ON at a moment's notice (much faster than passing from OFF to ON). Obviously, this mode comes with a trade-off, since more energy is consumed while keeping them in sleep mode than in OFF state. Gandhi et al. [212] studied the use of the sleep mode in datacenters. According to their work, servers are only busy between 10% to 30% of the time, despite virtualization. However, they utilize 60% of the total available energy. They show that utilizing the sleep state considerably reduces the energy consumption, especially in large datacenters.

Villebonnet et al. [213] use sleep modes to address the problem of energy proportionality in datacenters. According to the authors, the over-provisioning of resources in heterogeneous datacenters represent a high part of their wasted energy consumption. The authors propose the Big, Medium, Little (BML) infrastructure model as a way to reduce energy consumption when processing variable workloads. This model propose datacenters composed of heterogeneous resources, where computation can be migrated to servers that fit their needs. For example, placing computation of "small" resources liberates others to be set to sleep, consuming less energy.

Hardware improvements provide the ability of saving energy. Through the use of these improvements, it has been shown in the literature that a significant amount of energy can be saved. These hardware improvements can be mixed with software algorithms to enhance the reduction of energy consumption.

Software Exploiting Hardware Mechanisms for Improving Energy Efficiency

The most important software techniques used nowadays in Cloud computing for energy efficiency can be divided into 2 groups: optimizations within a server and optimizations within a cluster of servers. A cluster of servers is a set of computers which work together. A cluster can be treated as a single system for optimization purposes.

Server level optimizations aim at reducing the energy that a server consumes by making a better use of the server's resources. Main software optimizations within a server are the following.

Optimization of the Operating System: According to the literature, different Operating Systems have different impacts on energy consumption. In fact, according to Vahdat et al. [214], the use of a power-aware OS can save up to 50% of the power consumed by the RAM. In the same work, the authors determine that, using an energy aware OS, the energy consumed by servers on transmission can be reduced between 32% and 56%. This energy savings are achieved by compiling and configuring an operating system adapted to a certain infrastructure. For example, commercial servers do not normally make use of USB connectors. Thus, during the default initialization of the BIOS, the mounting of these ports is nothing but an unnecessary expense of resources. By avoiding this process, energy can be saved. On the other hand, up-to-date operating systems make a better use of modern hardware improvements, which increases the reduction in energy consumption.

Dynamic voltage and frequency scaling (DVFS): Voltage and frequency in the server's components can be scaled to reduce energy consumption. Howard et al. [215] propose the use of DVFS in processors to reduce energy consumption in servers. According to the authors, it can reduce th energy consumption in the CPU up to 80%.

Adjustment of the CPU load: The CPU is one the most consuming components of a server, according to the results of Fan et al. [216]. In an ideal environment, there is supposed to be a relation between the CPU load and the energy consumption, as shown in [198]. The discovery and setting of the optimal load of a CPU is an important part of the energy efficiency improvements. For example, Eads et al. [217] address the CPU load. The authors reduce the CPU load up to 69.95% and demonstrate that, through the reduction of load in the CPU, they are able to guarantee performance without a needlessly drain of energy.

Adjustment of hard-disk load: Virtual pages of inadequate size, non-ordered blocks of data and excessive accesses to the disk increase dramatically the energy expense of the hard-disk [205]. Due to the grouping of data and the principle of ordered location, each I/O reduces the overall energy consumption associated with the use of storage. This is because it is more probable to access a data block closer to the last one accessed than a far one, reducing unnecessary seeks [205]. Moreover, distributing the load in some disks, based on block popularity, allows setting the rest in power save mode [218].

Cluster level optimizations can be made so that the overall consumption of a cluster is reduced. The main difference with the previous type of optimizations is that, while optimizations on a server reduce the consumption of a server, in a cluster energy is often saved by increasing the consumption of a specific server (and reducing it in the rest). The most relevant optimization in a cluster are the following ones.

Consolidation algorithms: Consolidation is the process of placing the computation in the least number of servers. Virtual Machines can be migrated across different servers to increase occupation of active servers, freeing the rest of resources. By redistributing Virtual Machines in the cluster, freed servers can be switched OFF or placed in sleep mode. The optimality of the placement of the VM in a specific server in the cluster for energy efficiency reasons might be dependent on many environmental factors, such as external temperature or load of the server, as shown by Orgerie et al. [198]. As shown by Torres et al. [219] and Orgerie et al. [198] it is important to take into account the hypervisor's use of resources and the frequency of live migrations in consolidation algorithms. Other authors such as Hsu et al. [220] aim at keeping a high utilization of resources, but not maximizing it. In the work, the authors propose consolidating VMs so that every virtual machine uses less than 70% of the available CPU power in the server. Through this technique, the authors keep the CPUs occupied, but reduce energy consumption of the CPU when it is used over 70%, which the authors claim dramatically increases energy consumption. Their simulated results show an improvement in energy efficiency of up to 17% compared to resource maximization.

Placement algorithms: These algorithms tackle the distribution of VMs in the cluster on arrival. For example, when a new virtual machine is created, this VM can be placed in already active servers, if the server has enough free resources. This way, it is avoided switching ON new servers. New virtual machines can be also redirected to other active servers so that new virtual machines are consolidated in one or more servers. Then, when these servers finish their computation, they can be turned OFF, as shown by Orgerie et al. [120].

Server ON/OFF algorithms: These algorithms are based in the assumption that when a server is not being used, it is better to shut it completely down the server or some of its unused resources, such as Network Interface Cards or fans [120]. However, the energy costs of shutting a server down and awaking it again can be too much compared with the energy saved, in some cases [221]. If the idle time between computations is too short (i.e. the server was switched OFF and ON in a short period of time), this method may lead to an increase in energy consumption. Moreover, switching from an OFF state to a fully operational one takes some time [221]. While the server is being switched ON, the user needs to be waiting, which shows that the system is not responsive enough. To avoid this, datacenters keep fully operational servers unused to host incoming requests in cases of peaks of computation, which decreases energy efficiency. To avoid having active unused resources, idle states and arrival of computation can be predicted, so that it is possible to act consequently, as shown by Srivastava et al. [222]. In their work, the authors simulated the results of predicting idle periods of computation using real-life data. They showed that a significant amount of energy could be saved using prediction (up to 50% in some cases).

Server sleep algorithms: These algorithms are similar to the ON/OFF algorithms. However, switching from sleep to fully operational mode also has a lower response time than switching from an OFF to a fully operational mode [223]. Therefore, existing works focused on enhancing sleep mode algorithms take into account the shorter response time of devices compared to OFF-ON. Several works have studied the use of sleep mode on clusters of servers to enhance to reduce energy consumption, such as Krioukov et al. [2]. In their work, the authors simulate an average datacenter, and are able to save up to 63% of the normal energy consumption through the use of sleep algorithms. They also claim that, in the theoretical best case scenario, their algorithms would be able to save up to 90% of energy. Meisner et al. [224] propose a system for rapidly transitioning from high performance to sleep mode. In their work, the authors evaluate the CPU as the most consuming component while active (80 to 150 W). They compare their solution to DVFS and conclude that switching between states is more effective to save energy than varying the voltage.

2.5.2 Energy Efficiency Improvements for Networking Devices

Energy efficiency in networks has not been addressed in the literature as much as in datacenters. However, energy consumption in the network is not always efficient. For example, network routes are redundant for failure tolerance [225]. This implies that some networking devices are essentially unused for long periods of time. In this section, we discuss the existing network improvements towards energy efficiency. Two types of improvement to reduce energy efficiency are described in the network in literature:

- **Hardware improvements:** These improvements are implemented on the devices' hardware and/or the network design. They can be exploited by software improvements to reduce the overall energy consumption.

- **Software improvements:** These leverage physical and hardware mechanisms to reduce the energy consumption in the network.

The infrastructural improvements of the ISPs' datacenters have been previously addressed, as part of the infrastructural improvements of datacenters.

Hardware Improvements

Similarly to datacenters, the network is composed of a set of devices, which draw electricity for computing and communicating data. These devices can be routers and switches, but also gateways (an interface which connects two networks), bridges, hubs and repeaters. Their hardware capabilities can be improved to reduce energy consumption. Some improvements, such as network design for energy efficiency, consider the network as a system and reduce the overall energy consumption in it. In opposition, other solutions, such as sleeping devices, reduce energy consumption by individually saving energy in the nodes. The main hardware improvements in network devices are the following ones.

Rate adaptation: Network devices are normally composed by one or more semi-independent network cards. Each one, at the same time, controls different interfaces. These interfaces can be set to different performance communication rates (e.g. 10Mbps, 100Mbps, 1Gbps, etc). Every rate has a different power consumption. As expected, when the transmission rate increases it becomes more energy demanding. For example, Gunaratne et al. [226] measured the energy consumption of a Cisco Catalyst 2970 series Switch under different rates. The authors show that a link in the device at 100 Mbps adds 0.3 W of the total energy consumption. On the other hand, if the link works at 1 Gbps the amount of energy consumed is increased by 1.8 W. In a different work, Gunaratne et al. [62] propose the use of a rate adaptation protocol, called Adaptive Link Rate (ALR), in communication networks. This protocol changes the rate at which a port is communicating with the network. Through simulation of real-life traffic traces, they show that Ethernet links can operate at a lower data rate 80% of the time. Using the right policies to determine data rate changes, energy consumption in network cards can be significantly reduced. However, the authors also determine that this technique produces negligible increase in packet delay (less than 0.5 ms).

Sleep mode: Similarly to servers, network devices or some components can be placed in sleep mode, as proposed by Gupta and Singh. [45]. For example, it is possible to place interfaces in a sleep mode, and awake them when needed [227]. In such a mode, they consume significantly less energy than when fully operational. According to Eckermann [228], a processor in a networking device implementing the sleep mode can consume only 3% of what it consumes when fully active. Networking devices in sleep mode consume more energy than when switched OFF (as, when switched OFF, the devices consume a negligible amount of energy). According to Muhammad et al. [223] the energy consumption of sleeping devices in optical networks is close to 0W. On the other hand, networking devices in sleep mode are faster to turn back on operational mode when needed to than passing from an OFF to an ON state. According to Reviriego et al. [227], waking up an interface of a router can take as few as 4.48 microseconds. The amount of time spent while changing between states may affect network planning, as described in Bolla et al. [229]. In their work, the authors propose using the sleep mode in core networks to reduce energy consumption. They built a prototype and used synthetic traces to determine how devices (i.e. routers and switches) could be set to sleep mode. Their results show that, in a real-life network, energy could be reduced by up to 40%. The IEEE organization also dedicated a task force to reducing energy consumption in Ethernet technologies [230]. They defined a standard in Ethernet communications, called 802.3az. In this standard devices are awakened (pass from sleep mode to fully active mode) when receiving a burst. Then, devices transmit this burst as fast as possible to set themselves back to sleep. The authors show that a 82% of energy can be saved per link using their standard [231]. This project was finished, and the task force has not met since 2010.

Switch ON/OFF network equipment: Redundancy is used to reduce network failures. This is achieved by using multiple network devices for the same path or part of it. However, sometimes redundancy can be put aside for periods of time. For example when the devices connecting part of a path are underutilized, their redundant network devices can be dispensed without. In this case, the whole network device can be switched OFF to save energy when the redundant paths are not needed, similarly to servers. Chiaraviglio et al. [232] used this technique to save energy in networks. They demonstrate that it is possible to reduce the number of nodes used in today's Internet between 5% and 10%, depending on the heuristic used, while still providing the same Quality of Service. To determine the necessary number of network devices, they formulate the definition of the problem as switching off network devices and links while still guaranteeing full connectivity and maximum link utilization. The authors determine that the problem is NP-complete. To solve it, they focus on the minimization of the total energy consumption, and they set constraints in the connectivity and maximum links utilization. The authors complement their work with a reduction on active links, to maximize energy savings, without loosing Quality of Service.

Network equipment can be also partially switched OFF, by deactivating some components. For example, many state-of-the-art devices (routers and switches) have the option of switching OFF some of their network cards. This way, devices sending information through these nodes can redirect traffic to the interfaces managed by the active network cards (i.e. the active route). At the same time, a network card can switch OFF one or more interfaces, reducing energy consumption. Again, the devices sending information to this node should redirect their data through a different interface. Gunaratne et al. [226] determined that switching OFF an interface in a Cisco Catalyst 2970 series Switch could save between 0.3 W and 1.8 W to the total consumption. This is especially useful in redundant paths. Two nodes in a network may have two or more links connecting them, due to load balancing, or resiliency of service. If this redundancy is temporarily not needed because all the traffic can be managed through one path, or the path does not need to offer resiliency, the unnecessary paths can be switched OFF. To do so, the network interfaces in the routers and switches in the route are switched OFF.

Network Redesign: This approach tackles the energy consumption problem by designing energy-aware networks. For example, by limiting the packet processing that needs more energy to only a group of routers and links and using protocols which include the previous solutions (such as Switching OFF and ALR). Mellah and Sansó produced a survey on the existing solutions to reduce energy consumption in networks [233]. According to their work, taking into account power consumption when designing the system, the network and the protocols can make a significant difference in reducing energy consumption. For example, by designing redundant networks which implement ON/OFF mechanisms. In fact, Chabarek et al. [234] shown that, by designing a network from an energy efficiency point of view, energy consumption can be reduced by up to 65%. The authors do not directly address changes in latency or Quality of Service, but set themselves a constraint to provide a competitive service. This approach has been later taken by other works in bibliography, such as Guo and O'Farrell [235], who demonstrated, through simulations, that almost 80% of the energy consumed in cellular networks can be saved through redesign. In contrast to Chabarek et al., Guo and OŃarrell claim to offer equivalent Quality of Service whilst still reducing energy consumption.

Advances in hardware technologies: Some new communication technologies reduce the energy needed to transfer the data while providing a better Quality of Service for the end-user. An example of these improvements are optical communication technologies. Gladisch et al. [236] showed that fiber technologies consume less energy than electronic access networks. The authors claim that the use of new technologies, such as optical fiber, saves even larger quantities of energy than what is already saved by power management solutions, i.e. reduction of performance and switching ON/OFF. From a physical perspective, the use of Complementary Metal-Oxide-Semiconductor (CMOS) and new superconductors reduce the size of the chips, which reduces energy consumption, as shown by Papadimitriou et al. [237].

Reduction of the distance traveled: As shown by Gunartne et al. [226], the length of the cable connecting two devices affects the energy consumption of the network. Furthermore, The distance between the source and the destination may also affect the number of devices involved in the communication, i.e. number of hops. Thus, approaches which reduce the distance between the server and the client are able to save energy by reducing the number of hops involved in the communication process. This is the case of the work from Valancius et al. [55] and the project DISCOVERY [26], both described previously in this chapter.

Software Improvements

In this section we classify the algorithms that make use of hardware mechanisms to reduce energy consumption in the network. Software approaches have a broader view of the network plan and/or the expected utilization of the devices than the hardware ones. They use this information to save energy, by making use of the hardware improvements in one or more devices and to adopt coordinated approaches among several devices.

Algorithms to exploit Rate Adaptation: These approaches dynamically modify the performance of the nodes to save energy. They are based on the assumption that when a node reduces its link rate, it consumes less energy. Nedevschi et al. [238] studied ALR algorithms to reduce energy consumption. They propose an algorithm to balance energy efficiency and delay constraints. To develop their algorithm, they considered the three following constraints: a network device does not have knowledge about future packets arrivals, there is a fixed and discrete number of possible rates; and all packets arriving while switching rates are considered lost. With these assumptions, the authors use prediction of packet rate to create an algorithm which eliminates coordination across network devices, thus is amenable to be implemented in high-speed network devices. The authors show an average energy saving of about 5% with minimal added delay, with an average lower than 5ms.

Algorithms to exploit sleep mode: These approaches take advantage of sleeping improvements in the hardware to reduce energy consumption. Thus, setting some of the elements in the network to sleep implies a re-routing of data along the remaining ones. Nedevschi et al. [238] studied algorithms to reduce energy consumption by setting network devices to sleep. The authors model the problem of setting devices to sleep as a function based on three parameters: power drawn in sleep mode, time to transition between sleep and fully functional mode and amount of packets lost while sleeping. The amount of packets lost while sleeping depends on the mechanism used for invoking and exiting sleep states. In their work, the authors consider two mechanisms: timer-driven sleeping, where the network devices are programmed to sleep, and wake-on-arrival, where the device is started after sensing incoming traffic on their input ports. Using the timer-driven sleep mechanism, all packets received while the device is sleeping are lost. On the other hand, using wake-on-arrival the first trail of packets, i.e. these packets received while exiting the sleep state, are also lost. To solve the latter problem, the authors propose the use of dummy packets. For their algorithm, the authors consider a buffer-and-burst scheme. In this scheme, a router in the network buffers all the packets destined for the various next hops. After an interval, it sends all the packets in a "train of bursts" to every next hop one after another. The authors conclude that, for network devices which have low utilization, the use of their algorithm could save a significant amount of energy, while adding a small and controllable delay and packet loss. As an example, they claim that with a utilization of 10%, only using a buffering time of 5ms allows network devices to spend 60% of the time sleeping, adding a total 5ms delay. As a comparison to using ALR, the authors conclude that ALR saves more energy than sleeping in these network devices which implement Dynamic Voltage Scaling for rate adaptation, especially when the network devices have a light utilization. On the other hand, using frequency scaling alone seems to provide similar results in energy efficiency between sleep modes and ALR. The authors propose as a use case using ALR during day time, when the bandwidth utilization is higher, and passing to sleep modes during night time. Also, Muhammad et al. [223] showed the efficiency of sleep mode in redundant optical networks. Towards reduction of energy consumption, the authors consider 3 different algorithms to exploit sleep modes: *Minimum Power with devices in Sleep power*, *Minimum Power* and *Minimum Costs*. The first algorithm focuses on saving as much energy as possible when the device is set to sleep. This algorithm sets the devices to sleep as whenever they are not used. On the other hand, the strategy of Minimum Power focuses on saving energy in the overall process, including switching between states. That means that a device will not be set to sleep unless it is expected to remain sleeping for enough time. Finally, the strategy of Minimum Costs also tries to save energy using sleep mode, by reducing the overall monetary costs. The algorithm takes into account the overall costs of of the network and how using sleep mode affects. The authors determine that, at least, a 20% of links can be set to sleep, while meeting the expected Quality of Service.

Algorithms to exploit partial switch OFF in routers: These approaches make use of the improvements in hardware which allow switching OFF elements in the device, keeping the device operational at a lower performance. Raman et al. [239] studied the exploitation of switching on/off Ternary Content-Addressable Memory (TCAM) banks to save energy in networking devices. They proposed an algorithm which makes use of three different timers to select which banks to switch off. Two of these timers are used to delete different types of outdated entries. Then, when the timer reaches 0, the referenced bits are cleared. The third is used to compact the remaining entries. When this timer reaches 0, it triggers the algorithm to group the storage in the TCAM. While their algorithm increased the lookup delay, which affects performance, they show that by using parallel search this delay is avoided. Finally, they demonstrate, by simulation, that it can be saved up to a 33% of energy in a network device using their algorithm. Soteriou and Peh [240] studied the use of on/off techniques in network links. They showed that using a routing algorithm which adapts dynamically to the needs of the network, energy can be reduced by almost 38%. This energy saving is achieved through the derivation of the connectivity graph, i.e. shutting down all links which are not necessary. However, the authors found an increase in latency of about 2 ms in some cases. This increase in latency depends on the aggressiveness of the algorithm, i.e. the number of links shut down is proportional with the increased latency. In their experiments, Soteriou and Peh determined that by shutting down 2 links per router, the latency is increased by a factor close to 50%. This result is influenced by the traffic traces and the topology used.

Switch OFF algorithms: These algorithms plan the network topology so underutilized network devices can be switched OFF. According to Rost et al. [241], these can be flow-based algorithms, which center on the flow of data along the network and redistribute it between network devices, or destination-based algorithms, which center on the destination device and how to route the data through the network in the greenest way. According to the same authors [242], 50% of the network devices in the backbone networks are prone to be switched off, following their simulation results. The authors claim that more energy is saved by switching OFF network devices than switching OFF links, and provide a set of heuristic-based greedy algorithms. They compared four different heuristics to switch OFF network devices, with a constraint of reducing the number of network devices by iterating in a list of neighbor network devices. First, a random heuristic is proposed, where network devices are sorted randomly. Second, they used a heuristic where network devices are sorted increasingly, based on the number of links they have. After, they present a Least-Flow algorithm, where network devices are ordered increasingly by the amount of information flowing through them. Finally, they present an opt-edge algorithm, where network devices are first parsed to take out these that cannot be switched off (edge and aggregation network devices) and the rest follow. They show that opt-edge provides the most energy efficient solution.

Software Defined Networking (SDN): SDN is a technology which separates the control plane (where the route is computed) and the forwarding or data plane (where the data are sent by making use of the control plane computation) in different devices. By separating the control from the data plane, the former can be virtual, profiting from all virtualization techniques for optimization. For example, in an ISP's datacenter the control plane of several routers can be run by a single server. This solution extracts this computation, and consumption, from multiple routers and consolidates it in a server. Then, data plane can be managed by lighter devices, which consume less energy. Assefa and Ozkasap [243], study SDN approaches to save energy. They divide the solutions in four categories:

1. All traffic aware solutions which reduce energy consumption by taking into account traffic in the network devices, i.e. switching ON/OFF. They show that traffic aware approach achieves power savings of up to 50% during low load periods

2. Solutions which save energy by compacting data in memory, i.e. switching ON/OFF banks of memory. In some cases, these approaches can save up to 80% of the consumption of the component.

3. Solutions taking into account the topology of the network to dynamically determine where to place rules. For example, a networking device can have all forwarding rules to a certain location, thus aggregating traffic.

4. Solutions which are aware of the end host, i.e. the server running the virtual control plane, saving up to 7.8% of energy.

Traffic engineering: Traffic engineering algorithms reduce the load in a network, thus saving energy by simplifying the traffic requirements in the network. In dynamic environments network load varies in time. Network devices can be adapted depending on the amount of traffic which goes through a network in a specific moment. In most cases, when the network load surpasses a maximum threshold, set by the network administrator, the configuration of the network is changed to adapt to the new requirements. These changes may be the change of the performance profiles of the network devices, activating new routes or balancing the traffic through two or more redundant nodes. As explained before, activating higher performance modes and/or increasing the number of nodes in the network affects the energy consumption.

Traffic engineering algorithms make a smart planning of the path between nodes. They make use of statistical knowledge, information on the nature of traffic and measurements and simulations to optimize resource utilization and network performance [244]. Traffic engineering can be also used for reducing the energy consumption by making a utilization of resources which consume less energy.

Vasic and Kostic [245] studied traffic engineering algorithms for bandwidth intensive services in core networks. They describe the problem as NP-hard, and decide for an algorithm which spreads the load among multiple paths to leverage ALR and using sleep mode. To achieve this, each router publishes its link utilization for it to be accessed by the rest of nodes. Edge routers use this information to better distribute traffic. Assuming there exist two or more paths connecting 2 nodes, using this on-line information, routers can select which path to use for saving energy. The authors demonstrate that traffic engineering can reduce the number of active links by 21%, i.e. setting all unused links to sleep. Similarly, their simulated results show that 16% of the devices could be equally set to sleep mode. Overall, the authors report energy savings of 8%.

Amaldi et al. [246] also proposed a traffic engineering solution for energy efficiency. In their work, the authors use a Mixed-Integer Linear Programming (MILP) approach with an energy efficient heuristic. They adapt the discovery of the shortest path of OSPF [176] protocol to reduce network congestion by traffic engineering. By finding optimal routes between nodes, the authors are able to reduce the number of active networking devices and put to sleep unused ones. To achieve energy efficiency, they expand OSPF [176] with the Equal Cost Multi-Path (ECMP) rule, which distributes traffic demands to the devices having more outgoing links to the shortest path to destination. This way, devices with the least outgoing paths to destination are freed to be set to sleep. The process is divided in three stages:

1. A pre-processing stage, where the nodes and links which can be immediately switched OFF are identified.

2. The MILP algorithm, is used to find more network elements to switch OFF. This algorithm determines the shortest path for every route, respecting the utilization constraints.

3. The network is checked a third time, in order to find if other elements can be still switched OFF.

The authors show that through this system up to 50% of energy can be saved in the best cases.

Finally, Adnan et al. [247] describe a method for load balancing between distributed datacenters. In this work, the authors use a set of algorithms to distribute load between different datacenters. This way, when the system discovers a datacenter which offers a more energy efficient solution than the current one, computation is migrated to the new datacenter. The authors show that their system can save between 20% and 30% of energy. Also, Rifai et al. [248] studied traffic engineering algorithms from a green perspective. The authors deployed a real-life network, over which they used the Energy Aware Routing (EAR) algorithm. The authors show that, through this algorithm up to a 20% of links can be saved (shut down) compared to OSPF.

2.6 Conclusion

In this chapter we have focused on evaluating Cloud computing from two different points of view: Cloud applications and Cloud architectures. We have offered a taxonomy of existing Cloud applications and we have compared the two Cloud architectures (i.e. centralized and distributed). We have also shown the most relevant Cloud platforms run on these architectures.

To compare existing Clouds platforms we have focused on two relevant characteristics for ubiquitous Clouds: Quality of Service and energy consumption. We have shown that, when dealing with Quality of Service, the user considers low latency as a main characteristic. On the other hand, we showed that energy consumption in the Cloud represents a large contribution of energy consumption and CO_2 emissions worldwide. Furthermore, we showed that while energy consumption in datacenters has been largely studied, mechanisms to improve energy efficiency in the network have been neglected, while the network has a significant impact on energy consumption in the Cloud.

While the current Cloud computing trend began to move towards the ubiquity of computation, the main current Cloud platforms (both centralized and distributed) still confine the Cloud into datacenters, which are generally far from the client. Distance leads to increased energy consumption in the network, higher utilization of the broadband Wide Area Network - WAN - and poor client experience, especially for latency-critical applications. As a solution, distributed architectures try to locate computation among different sites. These distributed Cloud architectures which bring data closer to the final client are emerging as an alternative to centralized ones. We showed that distributed infrastructures provide a lower latency than centralized ones. Furthermore, we showed that distributed Clouds can provide a more energy efficient service than centralized ones, as the use of networking mechanisms such as the reduction of the distance traveled by the data can reduce energy consumption.

We have analyzed existing distributed Clouds and we have shown that a trend exists towards the allocation of computation near the user. Most relevant distributed Cloud platforms are edge Cloud and mobile Cloud computing. We show that, while both share common grounds, they also have many differences. The main difference is that mobile

Cloud is designed to offload computation of the mobile devices, while edge Cloud is designed to provide a service closer to the client, thus reducing latency between the client and the datacenter.

From all visions of mobile Cloud computing, we focused on a distributed approach based on the exploitation of unused resources in close by devices. We show that, through this approach, users obtain a more adaptable system, as the platform is dynamically adapted to the nearby devices forming the infrastructure. However, they lack the support of a static and reliable infrastructure provided by datacenters. On the other hand, other mobile Cloud visions use edge Cloud infrastructures to host the mobile Cloud. While this vision provides a service which is managed by infrastructure providers, thus is more reliable than the previous vision of mobile Cloud, it confines the Cloud to a static infrastructure. Moreover, edge Cloud may suffer issues of interoperability of Clouds (as different edge Clouds are managed by different infrastructure providers) and is less adaptable to the user (as the infrastructure cannot be dynamically changed).

Chapter 3

Microclouds

In this chapter, we present a distributed Cloud architecture. The goal of this architecture consists of providing a service closer to the end-user. It focuses on multiple user applications where the users are located geographically close to each other. We classify our approach according to the existing literature presented in Chapter 2 and compare it with other relevant architectures.

The remainder of this chapter is organized as follows. Section 3.1 shows the motivation for the development of our solution. Section 3.2 describes the objectives and challenges on the development of the architecture. The approach is presented in Section 3.3 and the architecture is studied in depth in Section 3.4. In Section 3.5 the problem of resource management in our architecture is tackled. A comparison of our architecture with relevant architectures in literature is presented in Section 3.6. Finally, Section 3.7 highlights the most relevant points of this chapter.

3.1 Motivation

As stated in Chapter 2, centralized Cloud architectures suffer certain drawbacks in terms of energy-efficiency and Quality of Service to the client. In Chapter 2, we also show that this situation is magnified in the case of applications where the data is exchanged between users located in a very specific area, such as the case of certain social networks or on-line document editing applications.

Besides, low latency is a crucial feature for a system hosting applications with time constraints [94]. Latency does not only affect response time, but also the consistency of data. For instance for on-line document editing, since users modify data at the same time, it is possible to suffer conflict of versions when several users have significantly different latencies. The relation between latency and consistency has been previously stated in literature. For example, Rajkumar [249] claims that latency "is of primary importance in real-time systems [for synchronization]". In the same line, Wang et al. [250] evaluated the relation between consistency and latency. They modeled this relation in the case of replication based machines and demonstrate that there exists a trade-off between both.

An example of such a real-time application is Google Drive [79]. While in this application the delay plays a not vital role, it affects the perception that the user has of the service. When two or more users concurrently modify data from a file, latency may cause users to see different versions of the same document. Then, the user seeing an outdated version may make changes which result incoherent with the updated one. This situation is considered in several works, such as [251], where the authors state that "[latency of document changes] in a cross-cloud collaboration is crucial to user experience, which means how long it takes [for the system] to get a document change to become perceivable for collaborators in other cloud collaborative editing services".

As shown by Delaney et al. [164] distributed systems provide a smaller latency than centralized ones. Similarly, distributed Cloud architectures can offer a better performance in terms of Quality of Service and energy efficiency than centralized architectures as shown by the authors such as Goiri et al. [14] and Vilnius et al. [55]. Different solutions have been developed in the last years towards bringing the Cloud closer to the end-user [148]. To this end, the most relevant solutions in literature are edge Clouds [56] and mobile Cloud computing [252]. According to state of the art [148], edge Clouds provide a service closer to the user by making use of existing edge infrastructures (i.e. the ISPs' infrastructure) to host replicated Virtual Machines. This way, information is processed in the edge and later propagated to the Cloud datacenter.

Mobile Cloud computing was born as a solution to offload computation of the mobile devices [35]. In a mobile Cloud, a Cloud is formed between static and mobile devices [35]. For example, a set of mobile devices can offload computation to one or more domestic servers. Contrary to edge Clouds, mobile Cloud computing does not rely on a specific vendor infrastructure [35]. Also, this solution does not need of replication of content between the edge and the central datacenter, as it is discussed in Chapter 2.

Finally, as explained in Chapter 2 (Section 2.2.3), mobile Cloud offload computation either in datacenters or in nearby devices. On the one hand, the use of datacenters adds latency to the system, as explained in Chapter 2. On the other hand, the use of a "peered" approach where computation is offloaded in close by devices lacks the advantages of datacenters such as having a reliable and powerful infrastructure which is always available.

Most works use edge Clouds to distribute content, such as the work from Valancius et al. [55]. However, keeping multiple active copies of the same Virtual Machine brings new issues in the case of real-time applications. For example, in order to keep consistency between active copies, it is needed to keep frequent communication between them [250]. As studied in Chapter 2, frequent communication between different devices may result heavy on the network if the data to send are large.

In Section 3.2 the objectives and challenges of this thesis are shown.

3.2 Objectives and Challenges

The objective of this chapter consists of designing a solution which offers a Cloud closer to the end-user for real-time services in very localized areas. In order to host real-time services, the proposed solution needs to provide a consistent framework, i.e. data shared among several users needs to be exchanged with a low latency. While the response time depends on the application, real-time systems normally require to answer within milliseconds, as stated by Hatley and Pirbhai [94]. For example, in the case of broadcast emissions a latency of more than 20 ms is already considered high [253].

The proposed solution will consist of a platform able to integrate in a Cloud all devices which are participating of the same service in a specific area. This means that this solution should not focus only in a specific network, such as edge Clouds, or restricting the Cloud to the computational power of private devices, such as it is done in mobile Clouds. Also, this solution should target the previously stated weaknesses of existing architectures: it should make an efficient use of energy; include the network both as part of the constraints and the solution of the problem, i.e. network awareness; and provide better Quality of Service to the user, in terms of latency, than centralized architectures.

In Section 3.3 we show our proposed solution targeting these objectives.

3.3 Proposed Approach

We propose GRaNADA, a distributed Platform-as-a-service (PaaS) Cloud. GRaNADA targets real-time applications with a low geographical distribution of users. To keep consistency, GRaNADA considers only one version of the data active at any time, i.e. only one Virtual Machine is accessed and modified by the users. Also, by keeping only one active copy of data, the use of the bandwidth is reduced. This is specially relevant in edge networks where the bandwidth is limited [254].

GRaNADA creates a distributed Cloud which intends to reduce the number of devices involved in the computation and communication. Energy can be saved in the network by shutting down (e.g. as it is recommended in Juniper's high availability design principles [255]) or downgrading (e.g. dynamic interfaces [256]) underutilized resources such as routers and switches, as explained in Chapter 2. At the same time, datacenters reduce their energy consumption because the computation is outsourced off the datacenters, thus allowing them to reduce active resources (e.g. servers).

Finally, due to the proximity of the user to the Cloud, distance traveled by data is reduced compared to centralized approaches. In consequence, it is expected that latency in communications is also reduced. Distance can be also reduced compared to edge Clouds. Thus, latency is also expected to be reduced due to a lesser number of hops between users.

Along with GRaNADA, we propose DEEPACC, a cloud-aware routing protocol which establishes connections between devices in the network based on two variables: energy consumed by the network devices using this connection and latency between them. The formal definition of GRaNADA and DEEPACC is shown in Section 3.4.

3.4 GRaNADA

GRaNADA (GReen Network-Aware clouD Architecture) is a Platform-as-a-Service architecture for Cloud applications which provides an elastic platform aware of the networks between users. It is able to geographically distribute the load on the network by exploiting unused computational power. To do so, GRaNADA creates a Cloud for every group of users collaborating on the same application. The computational load of the original Cloud is distributed across the users consuming the service, moving it away from the datacenters.

GRaNADA leverages computational power not only from standard private devices (such as desktops and laptops) but also from all available devices in the network which are capable of hosting computation. Following a fog computing approach described in Chapter 2, our solution integrates in the network private devices such as domestic routers or Cloud in the edge architectures (domestic servers) and public ones, such as ISP's routers or smart-city infrastructures. The common characteristic of all these components is that they share an application layer, and thus services can be run on top of them [257].

Figure 3.1 shows an example of a schema of interconnections in GRaNADA. GRaNADA outsources the computation off the datacenter in the form of a Virtual Machine host in the edge (laptop, domestic router, ISP router, ...). GRaNADA also includes a central location (i.e. a datacenter) for backup and management reasons. The functionalities of all devices depicted in Figure 3.1 are studied in Section 3.4.2.

The edge devices are multipurpose and normally less powerful than servers belonging to Cloud datacenters. To ease the computational weight in edge devices, GRaNADA divides computation into small virtual environments. That is, edge devices only run a reduced version of the Virtual Machine previously hosted by the datacenter. To address

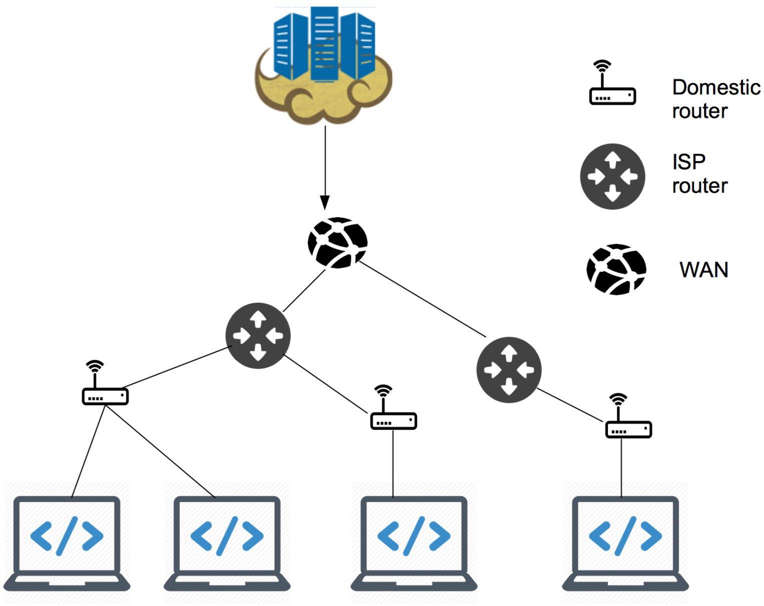


Figure 3.1: Schema of interconnections in GRaNADA

this division of computation, we define the concept of a *microcloud*, a small but functional version of the Cloud which hosts a reduced number of users.

3.4.1 Microclouds

A microcloud is an overlay network of devices which send and receive data related to a specific instance of an application. An instance of an application is a concrete occurrence of such an application. An instance hosts only data and functionalities of interest to one or more users who concurrently collaborate over the same environment. For example, an instance of a shared on-line document is formed by a file shared by several users and the associated functions, such as font style and size, dictionary in one language, etc.

To run instances, we define the concept of *Light Virtual Machine (LVM)*. A Light Virtual Machine is a type of operating system-level virtualization which hosts only the data and functionalities used in an instance of the application. A light Virtual Machine is less computationally demanding than the original Virtual Machine, due to the lesser amount of data and functions.

A microcloud is formed by these devices which either use the same instance of the application (i.e. clients) or are used to connect different clients (i.e. network devices). Figure 3.2 shows a schema of a microcloud making use of different devices. In this figure, microclouds are depicted as a group of devices drawn inside a cloud shape. This is a logical division, as a user may partake in different microclouds. For example, a user may collaborate on different documents at the same time. Inside a microcloud, each device takes, at least, one role: Provider, manager or client. These roles are studied in Section 3.4.2.

3.4.2 Roles in GRaNADA

In Figure 3.3, the internal functional schema of a microcloud is depicted. As shown, a node can participate in two different microclouds (client shown with double circle). We also include in our schema the possibility of one client connecting one or more other clients

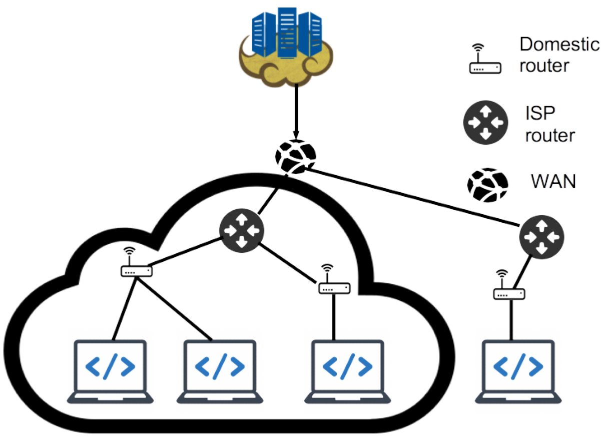


Figure 3.2: Representation of a microcloud from an infrastructure level

using an ad-hoc protocol. On the other hand, figure 3.4 represents a schema of microclouds from a platform level.

GRaNADA distinguishes three different roles between the members of a microcloud. Also, two additional roles are reserved to a central entity. Existing roles in GRaNADA are the following.

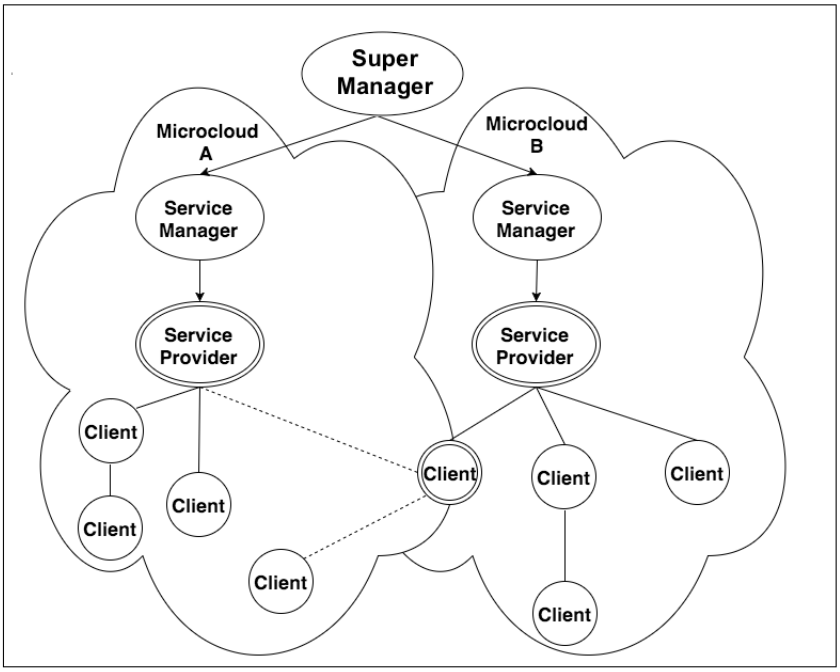


Figure 3.3: Schema of the roles in a microcloud

Base infrastructure: As it has been described before, GRaNADA makes use of a centralized infrastructure (i.e. a datacenter) for backup and management issues. This role is taken by a resilient and trustworthy infrastructure (such as a datacenter). The base infrastructure is used in the following cases:

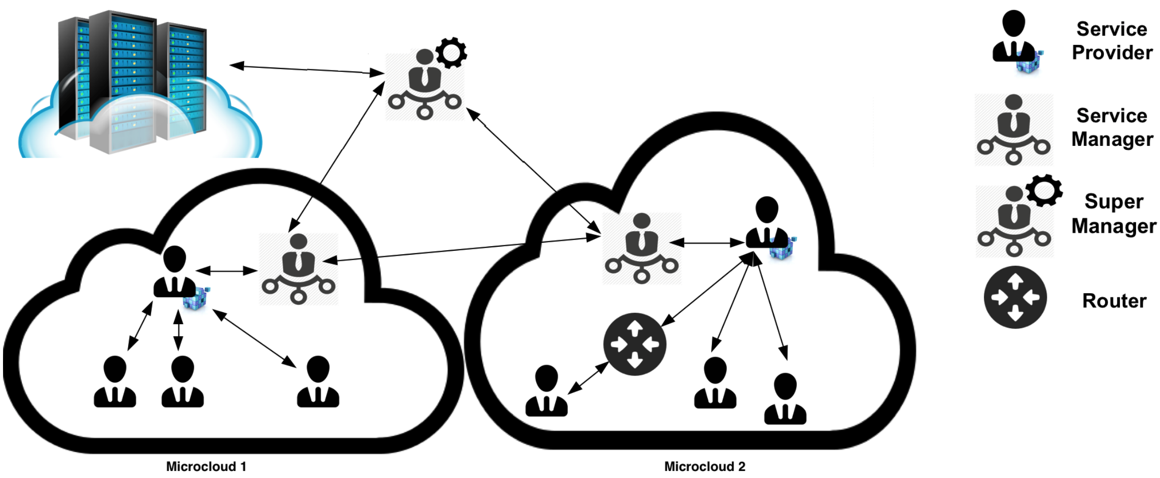


Figure 3.4: Representation of a microcloud from a platform level

To run the Light Virtual Machine if one or more LVMs cannot be hosted in the edge. When there is not enough available computational power in the edge, the computation can always be hosted in a base architecture. For example, an ISP edge datacenter can be used to host these Light Virtual Machines. Also, the edge datacenter can be used to host the LVM to reduce latency to all users, if it is placed in a location optimal for all users, such as on the shortest network path connecting two users.

To backup instances when they are not being used. When no user is using the LVM, it can be stored in the base infrastructure. Later, it can be accessed again from any user, independently of which device was hosting the LVM in the last session. It is also used for failure management. Asynchronously, GRaNADA copies the current state of the Light Virtual Machine and stores it in the base infrastructure. If the system fails, this image can be used to restore the LVM to a former state. Failure management is studied in depth in Section 3.5.6.

Super Manager: This role is given to a resilient, trustworthy and stable node, such as a server in an ISP datacenter. A super manager controls the creation and the topology of one or more microclouds, from an abstract perspective. When a microcloud needs to be created, the base infrastructure delegates this process to a super manager. Then, the super manager controls the initial topology of the system, and distributes the remaining roles to the nodes inside a microcloud.

Once the microcloud is created, the super manager guarantees that the microcloud offers, at least, a minimal Quality-of-Service to all nodes (set by the base infrastructure). To do so, the super manager periodically evaluates the QoS of the system. If the QoS is lower than the threshold set by the base infrastructure, the super manager changes the topology of the microcloud. These changes can be delegated to the service manager (detailed after) which can modify the existing internal topology or order a LVM migration inside the microcloud. The super manager can decide to merge or split one or more microclouds, as described in subsection 3.5.5. A super manager can control several microclouds through their respective managers. This is possible due to the hierarchical structure of GRaNADA.

Also, the super manager acts as a load balancer and provides a centralized management to its microclouds. Hence, decisions concerning the life cycle of microclouds controlled by the super manager can be taken. For example, when users want to join a specific instance of an application they request the address from the super manager. Here, a relation between microclouds and instances is kept. The super manager is, in this case, balancing the load in the system. Also, during the creation of a microcloud, i.e. when the first user launches an instance, it is controlled by the super manager. These situations are studied further in Section 3.5.

Client: The client role is given to any device consuming or contributing information. This role has no responsibility on the network. Before using the service, the client starts the join process described in Section 3.5.3.

Service Manager (SM): This role is assigned by the super manager depending on the nodes' computing capacity and robustness. Only one service manager exists per microcloud, and it is preferably assigned to a stable, static node, giving priority to static devices, such as laptops or domestic routers, over mobile ones, such as smartphones. The service manager is in charge of controlling the internal management inside a microcloud. The computation of the service manager is hosted also in a light virtualized environment (LVM). Its attributions are the following.

Placing the LVM inside a microcloud. The placement of the Light Virtual Machine affects the Quality of Service in two main directions. First, placing the LVM in a wrong device may cause interruptions of the service and/or poor performance. On the one hand, if the device where the LVM is placed is not reliable, it may fail, which causes service interruptions. On the other hand, if the device where the LVM is placed does not have enough unused computational power, the performance of the service may be deteriorated. Second, the location of the LVM affects the latency perceived by the clients. Placing a LVM at a very low latency from some specific clients may affect the latency received from others. As explained before, in real-time applications it is important not to have disparate latencies, to avoid consistency issues [164].

Manage failures of devices inside a microcloud. In order to keep a resilient service, the service manager reacts to failures such as failures in the device hosting the Light Virtual Machine. The reaction of GRaNADA to failures is studied in Section 3.5.6.

Establish the overlay network of a microcloud. The connections between devices are planned as to reduce latency and energy consumption. The service manager is in charge of planning the overlay network. It also reacts to devices leaving the network and modifies the overlay accordingly. The protocol used for the planning of this overlay is described in Section 3.5.2.

Also, the service manager balances load inside a microcloud. When a new device connects to a microcloud, it contacts the super manager. The super manager redirects its request to the appropriate service manager, which then includes the new device as part of the network. This process is explained in Section 3.5.3.

Service Provider (SP): The duty of the service provider consists in hosting the microcloud's LVM. This role is assigned by the service manager to a stable, static node, giving priority to static devices, such as laptops or domestic routers, over mobile ones, such as smartphones, and it is unique in the microcloud. The choice of the service provider is done based on the minimum delay to every client, reliability over time and hardware capacities. It is also preferably assigned to a stable, static node. When the service manager detects that a new node is a better fit for the service provider role, the role is assigned to the new node and the LVM migrated to it. The algorithm for selection of the service provider is described in detail in Section 3.5.

3.5 Resource Management

The management of resources inside a microcloud is carried out through the managers (both super manager and service manager). The software used for distributing microclouds in the system is run in the super manager. This software controls the creation, merge and split of microclouds.

The software for managing resources inside a microcloud is run in the service manager. This software is divided in two complementary protocols: management and routing protocol. Inside a microcloud, every device is assigned a role. This assignment is done through the management protocol. Every event, such as the initial formation of a microcloud, the addition or deletion of clients, a service failure from the provider or a microcloud division, trigger a different management algorithm. Through these algorithms, the service manager dynamically distributes the available resources. On the other hand, the routing protocol finds the most efficient route in between nodes, and creates a routing overlay network. These algorithms are detailed below.

3.5.1 Creation of a Microcloud

The process of creation of a microcloud is started in the super manager, when there exist enough unused resources to outsource computation. The first step consists in discovering the network devices connecting all the clients. To do so, the super manager asks every client to broadcast a message to all other clients to be connected to the microcloud. When this message arrives at a client, it carries information about the network devices on the path and their characteristics, such as energy consumption. Then, the client sends this information to the super manager. The protocol used to obtain the information in the path is described in Section 3.5.2.

Once the information about the network is obtained, Algorithm 1 is launched. It aims at building the network overlay that will be used by the microcloud clients in order to communicate and to use the microcloud services. This algorithm starts with a set of nodes which represents all the clients and network devices to be connected. In this set, at the beginning of the algorithm, every client is considered as a microcloud of only one device. The algorithm uses an A* algorithm with heuristic to find the best connection between any two microclouds. In each loop, two microclouds are extracted from the set. The shortest path between any node (either client or network device) is found. Then, this path is used to connect them, creating a new microcloud. The new microcloud is inserted in the path in exchange for the two previous ones. This process is repeated until only one microcloud is left in the set. The remaining microcloud is the result of the algorithm.

$$\begin{aligned} EC\ Associated\ to\ Connection_C = \\ (\gamma *\ Configuration_C + (1-\gamma) *\ Delay_C) \end{aligned} \tag{3.1}$$

Equation 3.1 shows our function of suitability for connections used in Algorithm 1 to build the network overlay. This suitability is calculated based on two factors: energy consumption of the path and delay between the two furthest nodes in the path. γ is used to favor energy efficiency (when $0 \leq \gamma \leq 0.5$) or distance (when $0.5 \leq \gamma \leq 1$).

Algorithm 1 builds the network overlay and decides which nodes take part in the microcloud (clients and some of the network devices in between them). In the following, we consider only these nodes. Once the microcloud is formed, the role of service provider is assigned. To do this, the suitability of placing the LVM in every node (client or network

Algorithm 1 Microcloud initial formation: merge function.

Put all nodes in $MicrocloudsList$ (as singleton microclouds).
if $size(MicrocloudsList) = 1$ **then**
  return the first element of $MicrocloudsList$
Remove $mc1$ the first element of $MicrocloudsList$
Remove $mc2$ a random element of $MicrocloudsList$
{*Connects the two microclouds using the most adequate connection*}
for each possible connection between $mc1$ and $mc2$ **do**
  Estimate the suitability of this connection
  if EC_new connection < EC_best connection **then**
    best connection = new connection
Create *mc_merge* as the microcloud made of $mc1$, $mc2$ and the best connection
Put *mc_merge* at the beginning of $MicrocloudsList$
return merge($MicrocloudsList$)

device) is calculated. The suitability of the placement is calculated according to two different characteristics: energy efficiency and Quality of Service. We gave special importance to latency as part of the Quality of Service. For instance, placing the LVM in a specific node may reduce the energy consumption (because the node is a very efficient one) but provide a high latency to some nodes. Then, both energy efficiency and latency need to be balanced. The provider role is assigned to the most fitting node. This process can be computed in parallel for every node following Dijkstra algorithm [258], thus reducing computation time, as shown in Algorithm 2.

Algorithm 2 Assignation of a provider.

Put all nodes in $Microcloud$ in $NodesList$
{*Calculate suitability*}
for each P in $NodesList$ **do**
  Compute Provider_suitability(P)
  if Provider_suitability(P) > Best Suitability **then**
    Best provider = P
return Best provider

Equation 3.2 shows our function of suitability used in Algorithm 2 to choose the provider. α is a configuration parameter used to favor energy efficiency (when $0 \leq \alpha \leq 0.5$) or quality of service (when $0.5 \leq \alpha \leq 1$).

$$\begin{gathered} Suitability\ Associated\ to\ Provider_P = \\ (\alpha * \sum_i Configuration_{P,i} + (1-\alpha) * \sum_i Delay_{P,i}) * Hardware\ Capabilities_P \end{gathered} \tag{3.2}$$

where:

- **Configuration** represents the energy efficiency of the device P when the minimum configuration possible, while managing the load, for every interface;
- **Delay** counts the latency between the possible service provider (P) and the node (i);

- and **Hardware capabilities** represents the capacity of a node to host the LVM, such as spare memory and CPU.

To retrieve the characteristics of each device in the path, GRaNADA uses a specific protocol, called DEEPACC. Devices in a microcloud intercept messages sent with this protocol and add their own information. DEEPACC is described in Section 3.5.2.

3.5.2 DEEPACC

The DEEPACC (Dynamic dEtection of Efficient Paths for Adaptive Cloud Computing) routing protocol is used to compute the most suitable route between every client and the service provider. A message under this protocol is sent always from the client to the service provider. Since the client does not know the address of the service provider, this message is broadcast as a discovery message to all neighbors following a flooding process. In every device in the path the message is captured, processed and updated with information such as the current Round-Trip Time (RTT), the current node's ID and its energy consumption. Then, the node forwards the message to all its neighbors. On the other hand, if a device finds itself in the message it assumes that it already sent this message and discards it. This is done to avoid loops. At the end of the process, the service provider obtains updated information about all routes between the client and itself. This information is later sent to the service manager, to plan the overlay topology.

Once created, the topology of a microcloud may change dynamically. These changes are caused when a new client connects to the microcloud, a device leaves the microcloud (for example when the client leaves the microcloud) or a network connection is broken (for example, due to a failure in a network device or a movement of a node). The process of joining (new clients) and detaching (node leaving) from a microcloud is described in Section 3.5.3.

3.5.3 Join and Detach Processes in a Microcloud

The processes of join and detach of devices are run respectively every time that a device connects or disconnects from a microcloud.

Join/rejoin process is executed by the client. When users want to use a specific service, they chose this service from a list, using a Graphical User Interface (GUI). This list is stored in the super manager, which has a static IP. Once the user selects the service to use, the client service joins a microcloud. Also, if the client experiences failures in communications (it cannot reach the service provider), it automatically rejoins the microcloud. In both cases, the same process is run: a joining process. As depicted in Figure 3.5, a joining process is divided in three steps:

First, the client obtains the address of the service manager through a request to the super manager. After that, a join request is sent towards the service manager, which answers with the address of the service provider. Then, the service manager has to obtain all possible routes between the client and the service provider. To do so, the client broadcasts a message to the service provider using the DEEPACC protocol described in Section 3.5.2.

Secondly, once the service provider obtains all the possible paths between itself and the new client, they are sent to the service manager. The service manager uses a Branch and Bound algorithm described in Section 3 to plan the microcloud's topology. After, the service manager communicates the modifications in the overlay topology to the service provider and to the super manager. Also, it updates the information about physical connections in the network and shares this information with the super manager. Information about the possible physical connections is used by the super manager to reconfigure microclouds, as explained in Section 3.5.4.

Finally, the service provider sends an acceptance message to the client along the chosen route. Before forwarding the message, every node (either network device or ad-hoc client) in the route, updates its routing table to the service provider.

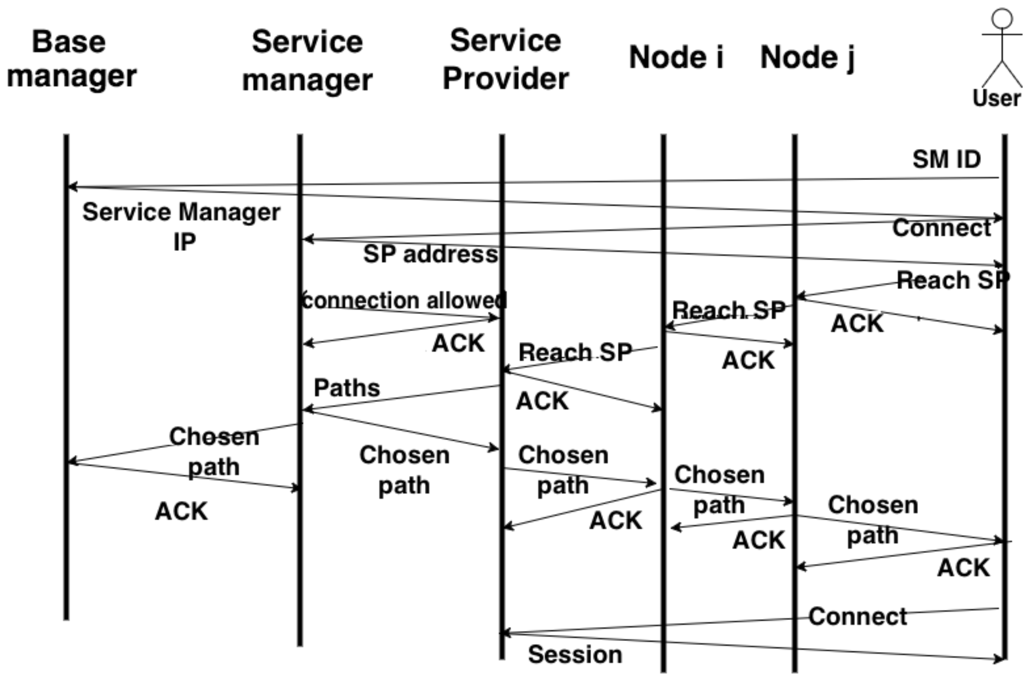


Figure 3.5: Joining a microcloud

When the service manager determines that one client has left the microcloud, it launches the **Detachment process**. This process is depicted in Figure 3.6. The detachment process is divided in three steps:

First, the service manager ensures that the client has left the microcloud. When a node is detached from the microcloud, the topology of the overlay network changes. A node detaching from the network can be a client or a network device. A client detaches from the network either by active disconnection (users log off their session) or because of failures in the device. When a client disconnects from a microcloud it sends a disconnection message to the service manager. Failures are discovered by the service manager. The node acting as a service manager sends a periodic message (heartbeat message) to all the clients in the network. If this message is not answered, then the client gets flagged. If the client gets several flags, then it is considered as dead. Every time a client answers a management message all flags to this client are cleared. When a client is marked as dead, the process of detachment from the microcloud is started. Besides, a network device is always detached in case of failure. A failure in a network device causes a disconnection of all the clients connecting through it. The process of detachment of nodes is started by the service manager, once it cannot contact the clients. On the other hand, if the connection between a client and the service provider is broken because of a failure in the network, the client will start a new addition process to try to reconnect using a different path.

Secondly, once the service manager has confirmed the detachment, it launches Algorithm 4. This algorithm checks the path between the detached node and the service provider, and eliminates from the overlay network those devices which are not used (i.e. network devices which were included to connect the detached client). The goal of this algorithm is cleaning the overlay network of unused nodes.

Finally, it communicates the changes in the topology to the super manager. The super manager updates this information for future reference.

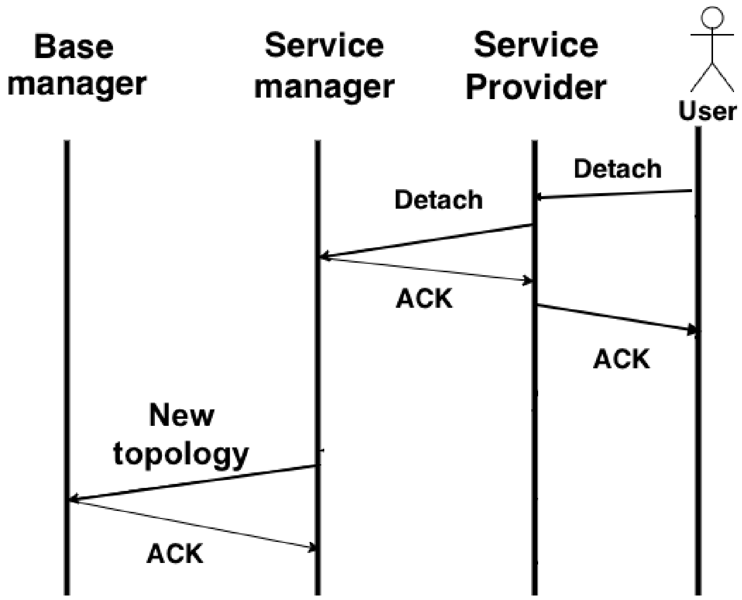


Figure 3.6: Detachment from a microcloud

Algorithm 3 is run every time a node joins a microcloud. This algorithm finds the best connection (in terms of energy and latency) between the new client and the microcloud by the service manager. To do so, the service manager checks all possible paths connecting the new client to any node in the microcloud. Paths are evaluated using Equation 3.1 and the most suitable one is added to the overlay.

Algorithm 3 Addition of a node to a microcloud.

for each node n of the microcloud **do**
  if Suitability_new connection > Suitability_best connection **then**
    best connection = new connection
Create *mc_merge* as the microcloud made of the initial microcloud, the node n and best connection
return *mc_merge*

Algorithm 4 is run when a node is deleted form a microcloud. The algorithm erases the client from the overlay. Then, the service manager checks the next hop in the path connecting the client to the service provider. If the next hop connecting the client to the service provider is a network device with only one neighbor (that is, if the network device was included in the microcloud only to connect this client to the service provider), the network device is erased from the microcloud. This process continues until reaching the service provider, a client (assuming the deleted client was using a different client's device to connect ad-hoc) or a network device with more than one neighbor (implying this network device is used to connect more than one client).

Algorithm 4 Deletion of a node from a microcloud.

Put all nodes in the path between the node to delete and $Service_{Provider}$ in $NodesList$.

for each n in $NodesList$ **do**
  if n is not a client AND n is not a provider AND n is not a manager AND n has only one neighbor **then**
    remove n from the microcloud
return microcloud

Finally, if a microcloud becomes too large to be managed, it may be split in two smaller microclouds. In a similar manner, when two microclouds share enough resources, they may be merged in one large microcloud. Both processes are described in Section 3.5.4.

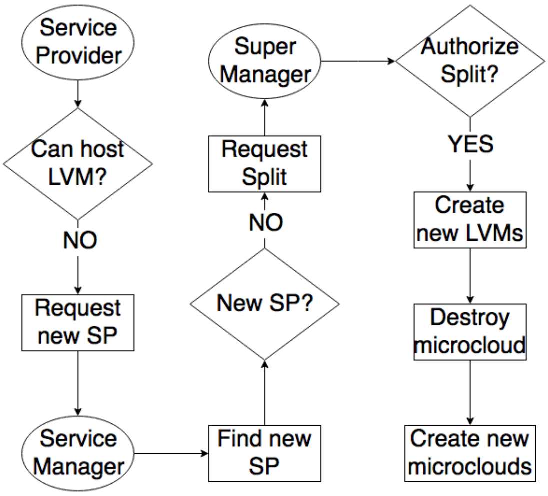
3.5.4 Merge and Split Processes

Both split and merging processes are controlled by the super manager. These processes are used to provide a better Quality of Service and reduce energy consumption. These process are based on adapting each microcloud to its dynamic needs. A microcloud can be divided into two or more microclouds (split) or several microclouds can join efforts into a single one (merge).

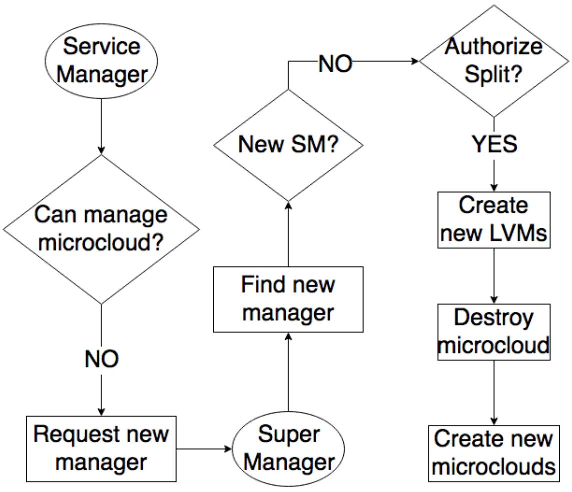
Splitting a microcloud in two is useful when the Light Virtual Machine becomes too big (i.e. the LVM requires more resources than these available in any node) and when the amount of nodes in the microcloud is unmanageable (i.e. the processing required by the manager is bigger than what any node can spare). By splitting a microcloud, the computation for hosting the Light Virtual Machine (service provider) and managing the network (service manager) are divided in smaller chunks. This way, the use of resources in the devices hosting these roles is reduced. When a microcloud is split, two different LVMs hosting information for the new users are created. A microcloud can be split only if there are two distinct groups of users (i.e. two different instances can be created which are not cloned).

The splitting process can be started by the service manager. There exist two cases when a microcloud is split: there is no available service provider which can host the LVM or there is no available service manager which can control the network. Figure 3.7a describes the process of splitting a microcloud when the LVM becomes too big for the SP. As described in this figure, if the service provider (SP) determines that it cannot host a LVM, it sends a message to the service manager requesting to drop the LVM (so the service manager assigns it to a different device). If the service manager cannot find a device which can host the LVM, it requests a split from the super manager. If the super manager authorizes the split, it determines how to split existing nodes into different sets (at least two) to form new microclouds. Then three steps are necessary: creating the new LVMs, destroying the former microcloud, and create two or more new microclouds for the new LVMs. To create the new microclouds, Algorithm 1 is run for each new service (i.e. for each LVM). After, the new service providers of each microcloud are selected using Algorithm 2.

Figure 3.7b describes the process of splitting a microcloud when it cannot be managed by the SM. This process is begun by the service manager, which requests from the super



(a) Flow chart of a split process caused by the SP



(b) Flow chart of a split process caused by the SM

Figure 3.7: Split process flow chart

manager to be eased off computation. If the super manager cannot find a new device with enough resources as to manage the microcloud, then it considers splitting the microcloud. If it can be split, the process of creation of new LVMs and microclouds begins.

As for the merging process, it is always triggered by the super manager. This process is executed when two or more microclouds are working on LVMs which have a strong interactivity (i.e. users of different microclouds are working on the same instance, but this instance is replicated in two or more LVMs). If the super manager considers that a LVM can be created blending those ones, the merging process is launched. First, the super manager creates the new LVM. Second, the super manager destroys the former microcloud. Finally, it creates the new microcloud. To do so, it executes Algorithm 1, using as an entry array all clients in both microclouds. After that, following the steps in the creation process, Algorithm 2 is used to find the new service provider. This way, a new microcloud is created independently from the distribution of nodes in the former microclouds. This ensures always having efficient connections inside the microcloud.

Along with the split and merging of microclouds, another approach to preserving the Quality of Service consists in migrating the Light Virtual Machine if needed. Dynamically, the system evaluates the Quality of Service to every client and may decide to migrate the LVM to a different device if, by doing this, a better Quality of Service is given. This process is described in Section 3.5.5.

3.5.5 Migration of LVMs

The system ensures a minimum Quality of Service for all clients. When the topology of the overlay network (i.e. the microcloud) changes, the location of the LVM may dynamically adapt. Topology changes may be caused by new clients joining or leaving the network; or changes in the connections between nodes (clients or network devices). Changes in the connections between nodes may be caused by the mobility of some of these nodes or by a reconfiguration of the topology. When a node moves in the network, it changes its neighbors (nodes to which it is directly connected).

On the other hand, both the super manager and the service manager may make changes to the topology of the overlay network (i.e. changes in the paths between devices). These changes are taken to reduce energy consumption and/or increase Quality of Service. In mobile scenarios (when multiple users connect to the microcloud using mobile devices), topology changes are frequent. For example, this is the case for connected cars sharing information when they enter a certain area.

The dynamic nature of mobile scenarios makes unrealistic an approach where all clients always find the best route to the service provider. The reconfiguration of a microcloud introduces an overhead in computation and time, which is used to plan the new topology (computation in the service manager) and migrate the LVM (time in which the service is not active). An excessive reconfiguration would affect the Quality of Service of the client, due to excessive migrations. Also, this computation would drain the resources of the device running the computation of the service manager. Given that, we opted for the use of efficiency heuristics. In our system, reconfigurations are launched only when the efficiency of the system (i.e. the difference between the best and worst Quality of Service provided) surpasses a threshold. This threshold is set experimentally to ensure a minimum efficiency. Two heuristics are employed.

Global optimization heuristic: Since every microcloud is independent, their best configuration may result in a global inefficiency. On the other hand, the super manager has information about every possible route in the system, as it is transmitted to it during the join process. Thus, the super manager accepts every configuration which provides, at least, a minimum efficiency. This efficiency is determined by the number of nodes connecting clients and the service provider, which are not clients of the microcloud, such as network infrastructure nodes, and by the size of the microclouds. If the super manager detects a configuration under the minimum efficiency threshold, it requests a redistribution of one or more microclouds. For example, if a device is part of two microclouds, both service managers may be competing to use it as a service provider. In this case, the super manager may request a reconfiguration in one of them to ensure a minimum global efficiency. Also, devices in two different microclouds may be heavily collaborating. Then, the merge process described in Section 3.5.4 can be run to increase Quality of Service to the users.

Local optimization heuristic: Inside a microcloud, the Service Manager accepts every configuration providing a minimum efficiency in terms of clients' Quality of Service.

This way, the microcloud is formed faster than if the service manager searches for the optimal path. Also, it requires less computational effort from the service manager. On the other hand, some clients may be added and deleted to the overlay network and the physical network may be changed by failure of network devices or mobility of nodes. Using a local heuristic ensures that the number of reconfigurations is contained in this case. If this threshold is surpassed, then a redistribution - new routes or Service Provider migration - is requested inside the microcloud.

Finally, changes in the physical network may also impact the Quality of Service. In this case, failures in the devices affect users in different ways. Failure management is explored in Section 3.5.6.

3.5.6 Failure Management

Four different roles may fail in our system: super manager, service manager, service provider and client.

First, we contemplate the case of failure in the super manager. As explained before, the role of the super manager is taken by a resilient and trustworthy device. However, this device can also fail. A failure in a super manager would not stop the services already running, but the system would not be checked for inefficiency. Furthermore, no new service or node would be able to join while the super manager is unavailable. Also, the rejoin process may be affected, as the super manager is unavailable. In a rejoin process, a client needs to request access from the service manager. Since the super manager is not available, clients cannot retrieve the IP of the service manager. The clients may launch DEEPACC targeting their last known service manager. If responsive, the process can proceed. If not, the rejoin is impossible until the super manager is available. Traditional high availability and fault tolerance approaches can be used to reduce the probability of failures in the super manager and its down time.

Second, we study the case of a failure of the service manager. In this case, new users would not be able to join the microcloud, nor former users to rejoin it. Also, the internal inefficiency of the microcloud would go unchecked. To mitigate its effects, the service manager is replicated across different nodes inside the microcloud, called manager backup nodes, following a hierarchical structure transparent to the user. If the service manager is confirmed to have failed by the super manager, its role is taken by the next node in the hierarchy. It is the responsibility of the super manager to regularly check on the availability of the service managers.

In the case of a service provider failure, service would be completely stopped. To reduce the impact of such a circumstance, the role of service manager is also replicated in several nodes along with the LVM, following the same hierarchical architecture as defined for the service manager. A failure in the service provider is first noticed by the clients which cannot contact it. These clients send a message to the service manager, which also tries to contact the service provider. If there is no response, the failure management protocol is started. If the service provider is confirmed to have failed, then the service manager starts the next service provider in the hierarchy to take over. The LVM is totally replicated in the first device in the hierarchy, but partially replicated in these which are not probable to become service providers. Also, another copy of the LVM is kept in the base infrastructure as a backup. Even when no clients make use of a LVM, a working copy of the data always exists.

Finally, there might be a failure in the client. On the one hand, the client's path to the microcloud may be broken. A path to the service provider may be broken due to failure in one of the network devices or because the client was connecting using an ad-hoc connection to another client, which left the network. In any case, the client automatically starts the join process once observing that it has no connection. To address management of failure in networking devices, traditional high availability and fault tolerance approaches are used. On the other hand, the user's device may also fail, causing a failure in the client. This situation is common to all approaches and is outside the scope of this thesis.

Figure 3.8 depicts the most relevant types of failures which may happen inside a microcloud.

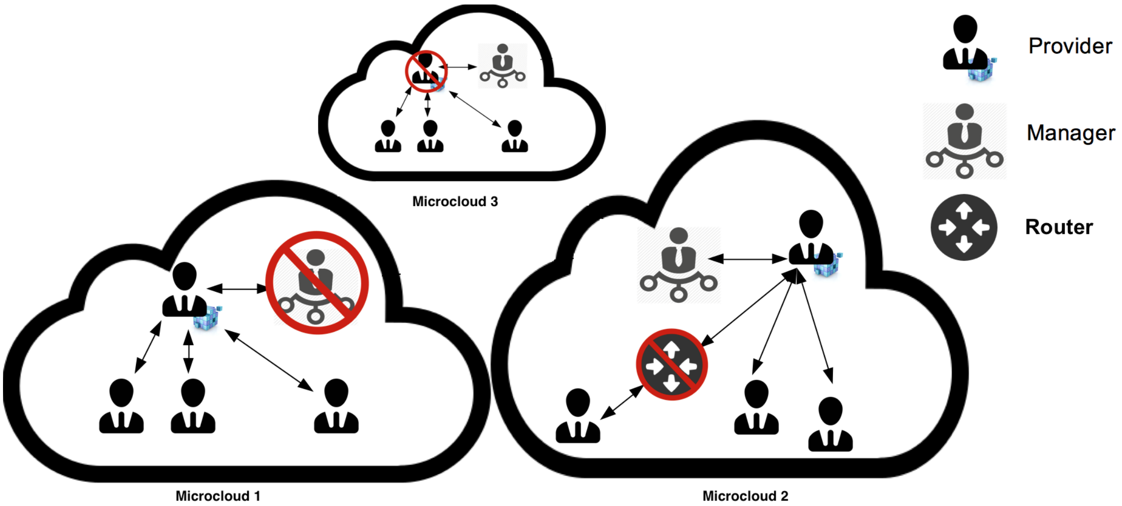


Figure 3.8: Failure of a Service Provider/ Service Manager/ Connecting node

3.6 GRaNADA vs Others

In this Section, we position our approach in the state of the art. As it has been stated before, GRaNADA follows a distributed architecture, so we will position it against most common distributed architectures. We have chosen a representative example of each one.

As an example of edge computing we have used NanoDatacenters [55]. The work from Valancius et al. is based on keeping a cached copy of the Virtual Machine in the users home gateway (ISP owned), to reduce latency and energy consumption. Data is regularly sent to the datacenter for consistency.

Also, we compare our approach against pure P2P architectures. As a representative of these architectures we used GNutella [259]. We have chosen this application for its relevance in the distributed field and the extensive literature on it. In GNutella, data are stored in the clients devices and replicated among those consuming them.

Table 3.1 positions our solution in the state of the art. As classifying characteristics, we have used the same ones as used in Chapter 2 (Table 2.2) to classify distributed solutions. As it is described in the table, GRaNADA's target is different than the other solutions. In the case of NanoDatacenters and GNutella, the main target is to distribute content, while GRaNADA aims at providing an architecture for real-time applications with concurrent modification from several users. Also, GRaNADA does not replicate active content and all users modify the same version of the data. On the other hand, all architectures share network and locality awareness. Finally, we add an extra characteristic: Energy awareness. It measures if the architecture is making efficient use of energy. While both GRaNADA and NanoDatacenters are energy aware, GNutella does not take into account energy while planning the system.

	GRaNADA	**NanoDatacenters**	**GNutella**
Target	Concurrent modification & Very localized applications	Content Delivery	Content Delivery
Replication	No active replication	Yes	Very high
Adaptability	Yes	Yes	Yes
Network orientation	Yes	Yes	Yes
Locality Awareness	Yes	Yes	Yes
Energy Aware	Yes	Yes	No
References	[64]	[55]	[259]

Table 3.1: Comparison of existing solutions

3.7 Conclusions

In this chapter, we have presented our approach towards building a Cloud closer to the user. We believe that the future of cloud computing benefits from a better distribution of resources. With that goal, we propose GRaNADA, a distributed PaaS architecture which distributes geographically the computation between the clients of the cloud to save energy and provide a better Quality of Service. GRaNADA is based on a novel concept of cloud computing, that we call microclouds. A microcloud is a set of computing and network devices which either work on the same instance of a service or are used to interconnect users of this service. This way, we isolate the utilization of a service to a reduced number of devices, in order to lower energy consumption and provide a better service for the final user. The use of microclouds gives the datacenter the ability of offloading computation in the network. To do so, the datacenter creates a copy of the Virtual Machine containing only the data and functionalities of interest to specific instance of the application. We call this a Light Virtual Machine (LVM). If the application is being used by several users to collaborate (for example, when several users share an on-line document), the LVM contains the data and functionalities necessary to provide the service only to the involved users.

In GRaNADA, the path between the clients is as short as possible, reducing the number of intermediary hops. Differently from other distributed approaches defined in Chapter 2, GRaNADA does not use multiple active VM copies. The computation, i.e. the Light Virtual Machine, is hosted uniquely in one device to which the other clients connect. This way, all users work on the same VM, copying data only for backup. Eliminating the use of multiple active copies of the Light Virtual Machine, i.e. users working on different copies of the LVM, the problems associated with distributed architectures which we defined in Chapter 2 (replication, network flooding, etc.) are avoided. Also, GRaNADA is network-aware, i.e. includes the network both as part of the problem and the solution. Finally, GRaNADA keeps a central point of management for the creation and destruction of microclouds, backups and failure management. By doing this, GRaNADA does not eliminate the improvements achieved through the use of centralized systems, but leverages the advantages of centralization in a distributed environment.

Chapter 4

Improving the Cloud's energy efficiency Adaptation of GRaNADA to a Core Network

In this chapter we present how GRaNADA is applied to a core network. The main goal of this adaptation of GRaNADA is to determine the energy efficiency of the architecture in a large scale use case. Through experimentation, we show that GRaNADA is able to save more energy than centralized and edge Clouds.

The remainder of this chapter is as follows. Section 4.1 shows the motivation of using GRaNADA to save energy and offers an evaluation on the distribution of energy consumption in a Cloud ecosystem. Section 4.2 describes how the architecture is implemented in the core network use case. An evaluation of our solution is shown and discussed in Section 4.3. Finally, the main conclusions are highlighted in Section 4.5.

4.1 Introduction

As shown in Chapter 2, the number and dispersion of devices connected to the Internet is expected to boom in the near future. The proliferation of clients negatively affects the energy consumption in datacenters [183]. In fact, the energy consumption in datacenters has been increasing in the last few years [25].

One of the most important consequences of increased energy consumption is climate change [260]. According to the Intergovernmental Panel on Climate Change (IPCC), the international body for assessing the science related to climate change, climate change is one of the greatest environmental problems that we will face this century [261]. Indeed, energy production and utilization accounts for two-thirds of the global greenhouse emissions [262]. Furthermore, energy consumption costs billions of dollars to Internet and datacenter providers every year [263, 264].

In this chapter, we evaluate the energy consumption of GRaNADA. For this evaluation, we model the energy consumed by Cloud architectures. To do so, we use a consumption model based on three infrastructures: datacenter infrastructure, intermediate infrastructure (network) and user infrastructure. Figure 4.1 shows the datacenter - client architecture and its distribution in energy consumption, as it was presented in Chapter 2.

As studied in Chapter 2, energy consumption in the datacenter has been already assessed in literature. For example, Krioukov et al. [2] analyze the distribution of energy consumption in a datacenter (servers, networking equipment, etc). Also, some other au-

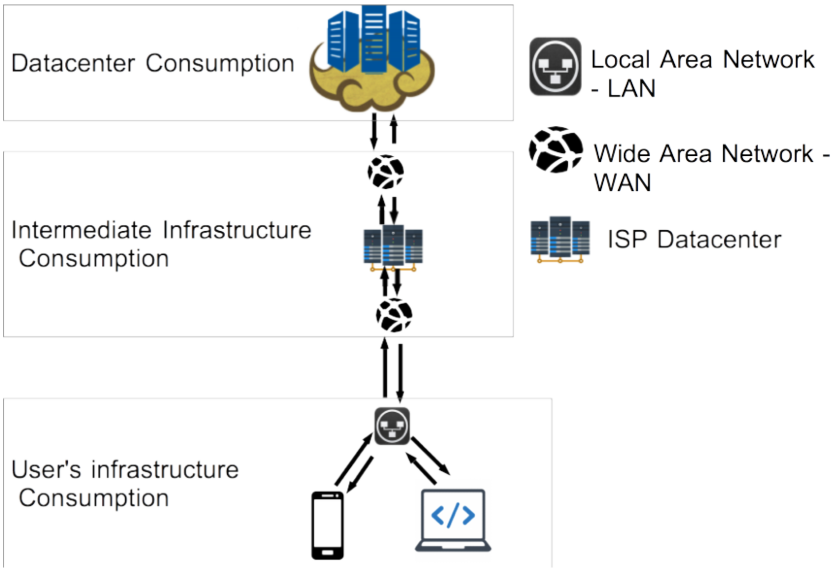


Figure 4.1: Datacenter - client consumption

thors studied energy consumption in datacenter networks, such as Orgerie et al. [190]. However, existing literature fails to provide a formal and simple model for the energy consumption of heterogeneous Cloud networks which fits our work.

We consider a Cloud network as the portion of bandwidth of existing networking devices used to connect any device client to the datacenter. We define this concept to determine what use a Cloud makes of external resources. As explained in Chapter 2, a networking device can be downgraded to save energy, or even switched OFF. However, changes in the performance of the network may negatively affect the Quality of Service provided to the Cloud user. To avoid this, the network adapts to the use of the Cloud, either by switching ON devices or upgrading its performance.

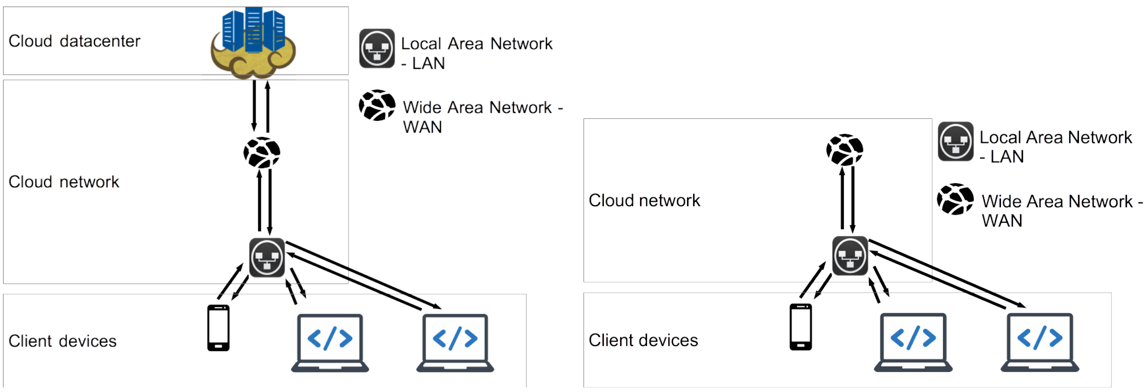
In the next section, we model the energy consumption of Cloud networks. Differently from other approaches, in our model we take into account the dynamism of Cloud systems.

4.1.1 Energy Consumption Modeling of Cloud Networks

Different Cloud architectures have different Cloud networks. For example, the Cloud network of centralized approaches is the portion of the network used to transmit data from each client to the centralized datacenter, as depicted in Figure 4.2a. This network includes both the WAN ISP network and the network at the user side, such as domestic routers, but excludes the network inside the datacenter. On the other hand, the Cloud network used in P2P-like Clouds is the portion of the network used to connect any two clients, as depicted in Figure 4.2b. In the case of GRaNADA, as described in Chapter 3, the Cloud network is the portion of the ISP network used to connect any client to the service provider, similar to P2P Clouds.

For simplicity's sake, we consider the Cloud network as an independent Cloud system. Every device in this network represents the new performance added to the network under the utilization of the Cloud. The characteristics of each device are determined by the difference between its performance forced by the Cloud and its performance without the Cloud traffic. For example, if a router needs to be upgraded to host the data generated by the Cloud, its energy consumption is represented in the Cloud network as the difference between current and former energy consumption.

Inside a Cloud network, several configurations are possible. Depending on the Quality of Service expected by each client, networking devices can use different configurations. For example, in the same datacenter-centralized Cloud network, a configuration can provide a certain Quality of Service to each client. If this Quality of Service needs to be improved, then the performance of the networking devices may need to be upgraded. An upgrade in its performance implies a higher energy consumption by the networking devices.



(a) Centralized Cloud Network (b) P2P/GRaNADA Cloud Network

Figure 4.2: Cloud Network

We define *Total Energy (TE)* as the energy consumed by all the networking devices in the Cloud network. The Total Energy is equivalent to the sum of the energy consumed by any path between the client and the virtual environment, called *Path Energy (PE)*. The Path Energy is calculated as the sum of the consumption of all devices needed to communicate between two devices. For example, the Total Energy of a Cloud network in a datacenter-centralized architecture is the sum of all the necessary connections between every client and the datacenter.

$$TE = \sum_{0}^{N} PE = \sum_{0}^{N} \sum_{0}^{i} Energy_{device} \tag{4.1}$$

Equation 4.1 shows the equivalence between Total Energy and Path Energy. In this Equation, N represents the number of paths (i.e. connections between client and the virtual environment) and i represents the number of devices on each path. The energy consumption of a networking device is distributed in *Base Energy (BE)* and *Configuration Energy (CE)*. The Base Energy consumption is the energy needed to keep a device on, excluding any kind of activity. In a Cloud network, the Base Energy is the energy that the device consumes in the lowest configuration. The Base Energy of a device in the Cloud network can be triggered in only if the network needs to switch ON a device to host the new traffic. In this case, the Base Energy becomes the same as the energy that the device is consuming with all interfaces OFF.

On the other hand, the Configuration Energy measures the consumption added to the device under a certain configuration and traffic. In a Cloud network, the Configuration Energy represents the difference between the energy used by a new configuration, compared to a former one. For example, if a networking device needs to be upgraded to host the new traffic generated by the Cloud, its Configuration Energy is found by subtracting the energy consumption before the upgrade to the new one. This division has already been used in existing works, such as Gunaratne et al. [226].

As an example, it can be used for the datacenter-centralized architecture described in Figure 4.3. For illustrative purposes, we establish that, in order to keep the Quality of Service, the ISP shall switch ON one router and upgrade the performance of a switch. The router needs to be switched ON and two ports should be set to a 1Gbp transmission rate (one for incoming and one for outgoing traffic). For the case of the router, we suppose that the transmission rate of one port needs to be upgraded from 100Mbps to 1Gbps. In this example, the new Base Energy of the router is equal to the energy consumption of the router when switched ON with all ports OFF. On the other hand, its Configuration Energy is equal to the extra energy which it consumes when only one port is working at 1Gbps. In the case of the switch, its Base Energy remains zero, as it was already ON, and its Configuration Energy is equal to the extra energy used when the port is working at 1Gbps.

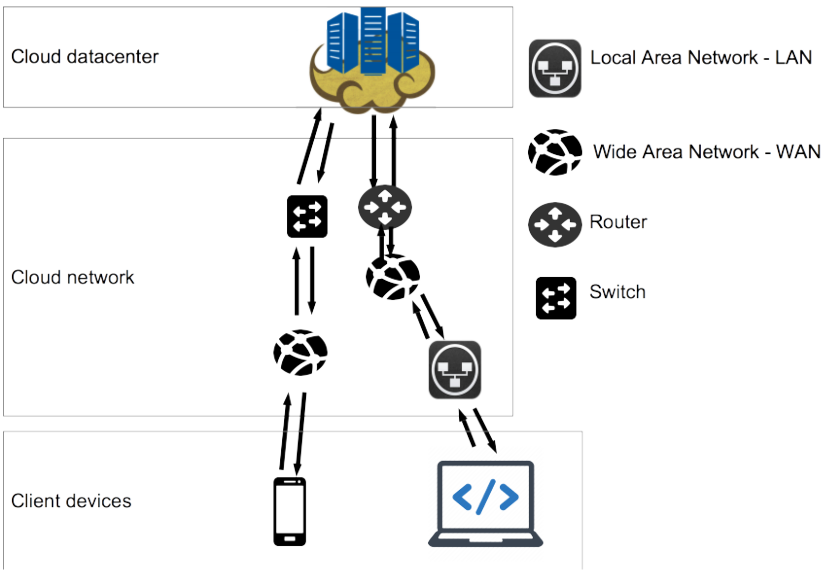


Figure 4.3: Datacenter - client consumption example

Concerning the parameters influencing the configuration consumption of networking devices, we focus on these addressing the configuration of ports:

- *Transmission rate:* As studied in Chapter 2, there is a direct relationship between the working rate of ports in a networking device (10Mbps, 100Mbps, 1Gbps, etc) and its energy consumption [62].

- *State:* This variable represents the state (ON/OFF/SLEEPING) of ports. As expected, different states have different consumptions [226].

- *Time:* We introduce a time variable to determine how long it is necessary to keep a port in a specific configuration to transmit some data. As studied in Chapter 2, some energy efficient solutions rely on setting ports to sleep or switch them off whenever they are not used [240]. Thus, these approaches may prefer using a more energy demanding configuration during a shorter period of time to set the port back to sleep.

- *Cable length:* Distance traveled by data in the cable has been shown to increase energy consumption [226].

- *Line cards:* The number of active components in the device also affects the energy consumption [229]. This means that line cards consume energy to stay active, even when all their ports are switched off. Thus, having the same number of ports active in different line cards consume more energy than having them in the same line card [234].

The rest of parameters - number of packets along the interface, IP protocol, etc. - are considered negligible [226]. Equation 4.2 represents the distribution of energy consumption in networking devices described above. The effects in consumption of the different parameters may differ from one device to the other. Because of this variability in energy consumption in different devices,Equation 4.3 describes the distribution of configuration consumption of a networking device as a function of the aforementioned variables.

$$\begin{aligned} Energy_{device} = Base\ Energy_{device} + \\ Configuration\ Energy_{device} \end{aligned} \tag{4.2}$$

$$\begin{aligned} Configuration\ Energy_{device} = \\ \mathbf{F}\Big(Transmission\ rate, state, Time, cable\ length, number\ of\ line\ cards\Big) \end{aligned} \tag{4.3}$$

Under the same configuration, the traffic load in a networking device has a negligible impact, as stated in several works [265]. However, it is possible to establish the relationship between energy and traffic, regarding how networking devices modify their configuration to allocate traffic under a certain Quality of Service. We address this value as *Traffic Energy*. Traffic Energy is the energy consumed under the minimum configuration needed to transmit all the messages (including headers, ACKs...) between two nodes, which respects the expected Quality of Service. Equation 4.4 defines the relation between the configuration energy and the traffic passing through the networking device at a given time.

$$\begin{aligned} ConfigurationEnergy_{network}(t) = \\ \sum[TrafficEnergy](t) \end{aligned} \tag{4.4}$$

4.1.2 Energy Optimization Issue

We define the Cloud network as the impact that the Cloud traffic has on the network. This impact should be minimized in order to reduce consumption in the network. From our model, the following implications are extracted:

First, Equation 4.1 shows that there is a relationship between the number of devices (repeaters, switches, routers, etc.) in a Cloud network and its energy consumption. This first situation can be tackled through the optimization of traffic. Traffic optimization can minimize the impact that the Cloud has on the network's energy consumption by reducing the number of networking devices in the Cloud network with a Base Energy higher than zero. When traffic is optimized, the Cloud uses the least consuming path to communicate with the clients, making an efficient use of networking resources. Then, unused networking devices can be either switched off or set in sleep mode, which reduces the energy consumption in the network. This situation can be exploited because network topologies are essentially redundant. This means that for every two main nodes in the network, there exists at least two redundant networking devices which can direct the same traffic [225]. This is, among other goals, used for traffic balancing when the traffic in the network reaches a certain load.

Second, according to Equation 4.2, different devices have different consumptions. For example, a 5 years old ISP router may consumes more energy than a state-of-the-art home switch, as has been studied in Chapter 2. This implies that energy efficiency does not only depend on the number of networking devices involved in the communication, but also how energy-efficient they are.

Third, according to Equation 4.3, two similar networking devices can have different consumptions depending on their configuration. A Cloud network may dynamically change its consumption, as the networking devices change their configuration. Thus, the energy consumption of a Cloud network is a dynamic value and has to be evaluated regularly.

Finally, Equation 4.4 shows that there exists an indirect relationship between the amount of data (load of traffic) in a Cloud network and the energy consumed. This relation is based on the different configuration profiles. The amount of data generated by a Cloud may affect the energy consumed by the network, as the networking devices have to upgrade their configurations to cope with the new data.

4.1.3 Objectives

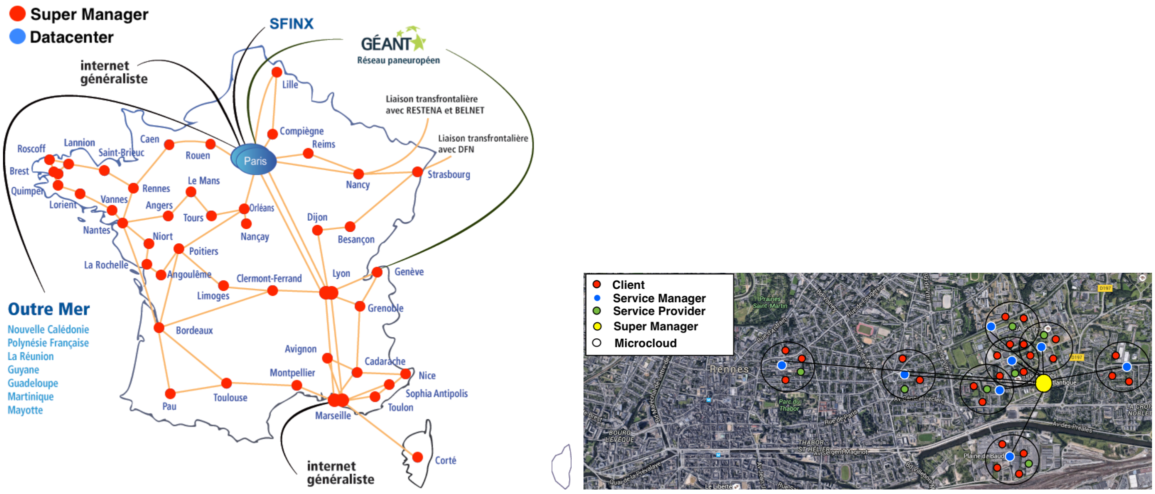
As it has been shown in the previous section, to reduce the energy consumed by the Cloud network, it is necessary to reduce the length of the path connecting two or more users and to make an efficient use of networking resources. The aim of this chapter is to evaluate the energy efficiency of GRaNADA. To make an efficient use of energy, GRaNADA uses microclouds to reduce distance between clients and the VM. That is, distance is reduced by connecting clients to the closest service provider. On the other hand, the use of the managers described in Chapter 3 ensures that the traffic in the network makes good use of resources. As an additional advantage, the computation hosted in the datacenter is diminished, as this computation is outsourced to edge devices. So the datacenter also reduces its energy consumption.

In Section 4.2, we show how GRaNADA is adapted to reduce energy consumption in a large-scale network. In this Chapter, we take the use case of a core network connecting several cities in the same country. As we showed in Chapter 2, the probability of two users sharing some data decreases when the distance between users increases. Thus, the motivation of adapting GRaNADA to a core network is that users inside a region share multiple interests. These interests may be set, for instance, by information mainly concerning inhabitants of this region (such as meteorological information, regional news or sports results) or translations of existing information into a regional language.

4.2 Adapting GRaNADA to a Core Network

A core backbone or core network is a Wide Area Network which offers "means of interconnection between the regional networks" [266]. In a core network, a Point-of-Presence (PoP) is an infrastructure that serves information inside a region [267], acting as a connection between the regional network and the core network. The infrastructure supporting this service is managed by the corresponding ISP [267]. For simplicity's sake, we assume that the datacenter providing the Cloud services is built near the node with the highest node degree (PoP with the largest number of direct neighbors).

A Point-of-Presence is a static and reliable node in the network. As determined in Chapter 3, such a node is prone to receive a management role. Also, as described before, it is expected that users in a region collaborate more often than users in distant regions. As explained in Chapter 3, when several users collaborate on the same data a microcloud is formed to host these data. Each microcloud is internally managed by a service manager, and a super manager controls several microclouds at the same time. Thus, in the case of a core network, a Point-of-Presence can be attributed the role of super manager.



(a) Description of GRaNADA adaptation to a core network (Renater network here)

(b) Description of GRaNADA adaptation inside a region (Source: GoogleMaps)

Figure 4.4: Adaptation of GRaNADA

Figure 4.4a shows the scheme of adaptation of GRaNADA inside a core network. Here, every Point-of-Presence acts as a super manager for a region. The datacenter is centralized in the node with the highest degree (Paris). When a microcloud is created, the correspondent super manager requests a Light Virtual Machine from the datacenter, and creates a microcloud inside the region. A more specific description of the adaptation of GRaNADA inside a region is shown in Figure 4.4b. Here, a representation of possible microclouds is shown. As explained in Chapter 3, every microcloud is managed by a service manager, which communicates with the super manager. The adaptation of a microcloud inside a region is studied in depth in Chapter 5.

In Section 4.3, we evaluate the energy consumption of different Cloud architectures, including GRaNADA, over the physical topology described in Figure 4.4a.

4.3 Evaluation: Energy Consumption of Microclouds

For our experiments, we simulated the French research backbone network *RENATER*. We used this network because it is representative of a core network and its topology is publicly accessible [268]. To increase the size of the network, we added two extra devices connected to each interface of the original PoPs. These new devices represent extra PoPs serving data to the region. We considered a central datacenter situated in Paris.

Rather than using a large number of nodes, in this Chapter, we opt for running a limited but accurate simulation. With this goal in mind, we used NS3 simulator [63]. NS3 is a packet level simulator, which provides accurate information about the communications. To provide an accurate simulation, NS3 reproduces the whole communication process up to a packet level. On top of NS3, we have used the ECOFEN module [190], which includes tools to evaluate the energy consumption of a network. Such a testbed has a large computational weight, since packet level simulators require large durations for computation. In fact, the computational costs of such simulations increase exponentially with the number of simulated nodes [269]. This tool gave us an accurate result for a topology including a total of 142 different nodes.

Since each node represents a Point-of-Presence, we have considered that each node is consuming as much as a single networking device. The consumption has been considered homogeneous (all nodes consume the same) because, in these experiments, the goal is to evaluate how distance affects energy consumption. As a value for the energy consumption of a networking device we have used the consumption of a DELL PowerConnect 6224 [270]. We have tested its energy consumption using a Power Distribution Unit from Server Technology [271], which incorporates a watt-meter. We carried out experiments on this setup to calculate the energy consumption of the device by switching OFF all ports and measuring its base energy consumption. After, we measured the consumed energy by switching ON ports to a 10Mbps rate and repeated this process for 100Mbps and 1Gbps.

According to our measurements, the base energy of such a device is 645 Watts. At full capacity (all ports ON at 1G), its energy consumption (base energy + highest configuration energy) is around 725 Watts. We have chosen to give each device in the network a consumption of 700 Watts (base energy + configuration energy), to provide an average value. While results shown in these experiments are dependent on the value given to the energy consumption, conclusions obtained from these results are independent of the consumption of the nodes. That is, even if the energy consumption of the nodes in the network in a different scenario may vary, the relation between the different approaches remains the same.

Given the lack of traces in literature for concurrent access to multiple users' Cloud applications, we have obtained several real 45-minutes traces from actual Google Drive sessions [272], using the network packet analyzer tool Wireshark [273]. These traces are obtained either from the concurrent use of documents (addition/deletion of text) or concurrent edition of pictures, such as applying filters, varying brightness, etc. In our simulation, PoPs exchange information using these traces.

We compare three different approaches in our experiments: a datacenter-centralized Cloud, a distributed approach with caching of data (edge Cloud), and GRaNADA. We compare these approaches in terms of energy consumption. We acknowledge the existence of other approaches not considered in this evaluation, such as P2P Cloud computing. However, either they have been ruled out in Chapter 2 due to their excessive energy consumption, as it is the case of P2P Clouds, or they are already represented by one of the solutions. For example, the case of distributed datacenters, can be extrapolated from the datacenter-centralized simulation.

For every approach, we minimize the Cloud network's energy consumption. That is, all networking devices are set to a performance as low as possible. To ease the comparison of approaches, we model the system as if the datacenter had a set of dedicated routers in the core network. That is, every node is represented as if it was a router which could either be switched OFF and ON according to the utilization of the datacenter. While in real case, this is not true (data addressed to a datacenter share the same networking devices as the rest), as explained in Section 4.1.1, the amount of traffic added by the Cloud affects the energy consumption of the network.

In detail, the compared approaches are:

All ON: In this approach, it is assumed that a client may connect at every instant, and the network is functioning full performance at all times. In this approach, the addition of a new node to the network is instantaneous, as the links are already active. However, it produces the least energy-aware system. This approach represents the datacenter-centralized architecture. For example, a datacenter in Paris is hosting the virtual machines accessed by the users.

SPO (Shortest Path Only): This experiment represents an edge Cloud architecture where the Virtual Machines are cached in the Points-of-Presence. In this experiment, Virtual Machines are sent to the datacenter after being modified, in order to keep consistency. The network maintains an open path between active nodes (i.e. Points-of-Presence which are hosting a cached VM) and the datacenter. To implement this approach we use a fully informed version of the OSPF protocol [176]. This protocol starts from a fully shut down network. Every time a node becomes active, the system calculates the shortest path between this node and the datacenter, producing the shortest path. Only nodes in this path are switched on, if they were not already. In this approach, for a node to become active it requires more time than the All ON experiment as some networking devices may need to be switched ON. On the other hand, it is more energy-aware. This approach represents architectures such as content distribution networks [128] where data can be concurrently modified.

DEEPACC: This experiment represents the adaptation of GRaNADA described above. To implement this approach, we use the DEEPACC protocol described in Chapter 3. This protocol also starts from a fully shut down network. Every time that a node becomes part of a microcloud, it calculates the shortest path between the sender and the closest node in the microcloud. Only the nodes in the resulting path are switched on, if they were not already. The main difference with former approaches is that nodes do not make use of the datacenter, but connect to the closest active microcloud. For the current experiment, we have considered the creation of only one microcloud. The use of multiple microclouds is left for Chapter 5. In this approach, when a node participates in a microcloud, all nodes between this node and the microcloud need to be switched ON. This is a longer process than the All ON one. On the other hand, DEEPACC provides a more energy-aware approach.

4.3.1 Energy consumption of the core network part

We performed two different sets of experiments. In our first set of experiments, we focused on comparing the energy consumption of the three different approaches when only two nodes are active. We considered this casuistic because in some cases the number of users of the network is low. For example, in a dedicated core network such as an enterprise core network, the load of data and users in the network is normally subject to the working hours of the participating centers (even though there are exceptions).

Figure 4.5 shows the energy consumption of the 3 approaches, where the 2 active nodes are 1 hop distant from each other. In the case of All ON, all the devices in the network are working and responsive, while none is used. Indeed, the energy consumption of All ON is static. That is, independently from the number of active nodes in the network, the number of nodes ON in the system is always the same (all). The result provided by All ON in this experiment acts as an asymptote for the other approaches. Neither SPO or DEEPACC will ever consume more than All ON. As it is shown in Figure 4.5, SPO consumes almost 90% less energy than All ON. Finally, DEEPACC is the protocol which consumes the least: up to 75% less energy than SPO. That is, a saving of 42 GWatts can be obtained in the best case scenario through the use of DEEPACC, compared to All ON.

These results were expected given that, since no other node is active, every node which is ON and not receiving data is being inefficient. In the case of the All ON approach, all nodes are ON and expecting future connections. All this consumption is being wasted.

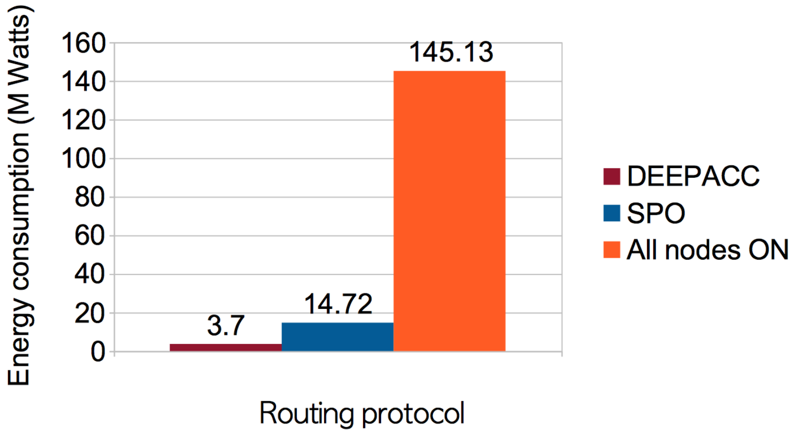


Figure 4.5: Energy consumption of DEEPACC when considering two clients close to each other

As depicted, the SPO approach performs much better. This is because it only keeps ON the path necessary for both users to connect to the datacenter at each moment. Finally, since DEEPACC does not need active communication at all times with the datacenter (only during backups), the LVM can be hosted in either one of the nodes. This way, once the LVM has been retrieved from the datacenter, all networking devices which are not connecting nodes in a microcloud can be shut down.

We acknowledge that in most networks and/or in a different time frame, the number of users is higher than the one considered for the first experiment, and its distribution variable. Thus, as a second experiment, we compare the evolution of energy consumption of the three approaches under an increasing number of active nodes. To carry out this experiment, we have chosen two activation scenarios: random and sequential. In a random activation scenario, nodes are chosen – to host a cache of the VM in the case of SPO or to be part of the microcloud in the case of DEEPACC – using a pseudo-aleatory algorithm. This means that the distance between the active nodes varies between experiments which activate the same number of nodes. In the random scenario, the distribution of active nodes along the network is high, having connection between distant nodes.

In the sequential scenario, all nodes are activated according to their proximity to other active nodes. This means that in all sequential experiments, the distance between active nodes is the smallest possible. This way, the users in the microcloud are concentrated in a geographical point and the distance between them is as small as possible.

For our first experiments, we chose a random activation of nodes. That is, in each loop, a random node is activated until as many nodes as desired are active. As expected, all inactive nodes are shutdown. Each experiment has been run 10 times to provide a statistically significant result.

Figure 4.6 shows the evolution in energy consumption of approaches SPO and DEEPACC when the number of active nodes increases following a random selection. In all the cases, DEEPACC presents a smaller energy consumption than SPO. This is explained by the nature of both approaches. On the one hand, DEEPACC reduces the set of active nodes to the shortest path between the client node and any other node in the microcloud. On the other hand, SPO reduces the active path to the set of nodes connecting every active node between them and at least one of the nodes to the datacenter. This means that the set of nodes activated by DEEPACC is always contained in the set of nodes activated by SPO. In the best scenario for SPO, the greenest path between two nodes in a

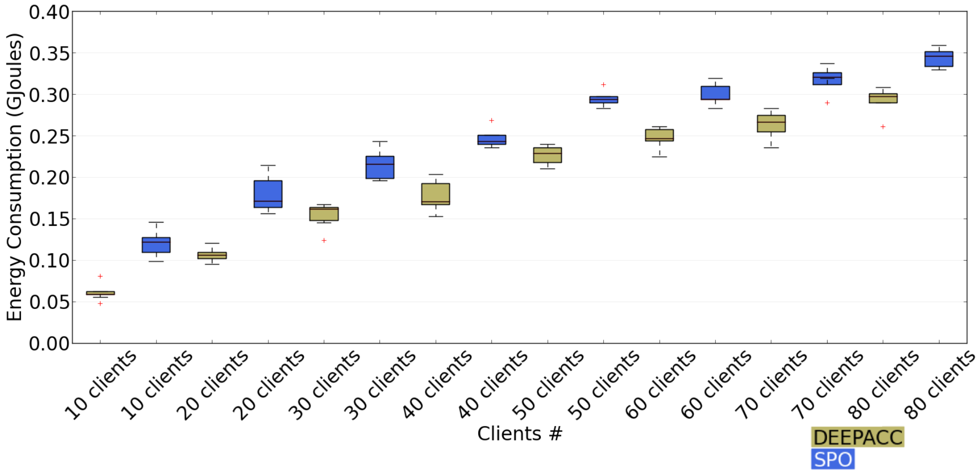


Figure 4.6: Random selection of active nodes

microcloud passes through the central node, which hosts the datacenter. In this case, the energy consumed by DEEPACC is exactly the same as the one consumed by SPO, as both paths pass through the central node. However, if the nodes are concentrated on one side of the network (i.e. there is no path which passes through the central node) DEEPACC always consumes less energy.

More formally, DEEPACC finds the greenest path between two nodes A and B. So, $DEEPACC_{Path}(A,B) = \{A, \dots, B\}$, that is, $DEEPACC_{Path}(A,B)$ is the set of nodes which forms the greenest path between nodes A and B. On the other hand, SPO finds the greenest path between nodes A and B to a central node C. Formally, $SPO_{Path}(A,C) = \{A, \dots, C\}$ and $SPO_{Path}(C,B) = \{C, \dots, B\}$. If both paths are merged, by appending the set of nodes $SPO_{Path}(A,C)$ to $SPO_{Path}(C,B)$, we get that SPO finds the greenest path between A and B, passing through C, formally defined in Equation 4.5.

$$SPO_{Path}(A,B) = \{A, \dots, C, \dots, B\} \tag{4.5}$$

We denote $DEEPACC_{Path}(A,X)$, the path with minimal consumption between A and X, where X is any node in $DEEPACC_{Path}(A,B)$. For $X = B$, $DEEPACC_{Path}(A,X) = DEEPACC_{Path}(A,B)$. For $X! = B$, $DEEPACC_{Path}(A,X) \subset DEEPACC_{Path}(A,B)$. That is, if every node in the path is part of the greenest path between A and B, then the path between A and any other node in the path needs to be greenest path between them, otherwise $DEEPACC_{Path}(A,B)$ would not be the greenest path. Thus, $DEEPACC_{Path}(A,X) \cup DEEPACC_{Path}(X,B) = DEEPACC_{Path}(A,B)$.

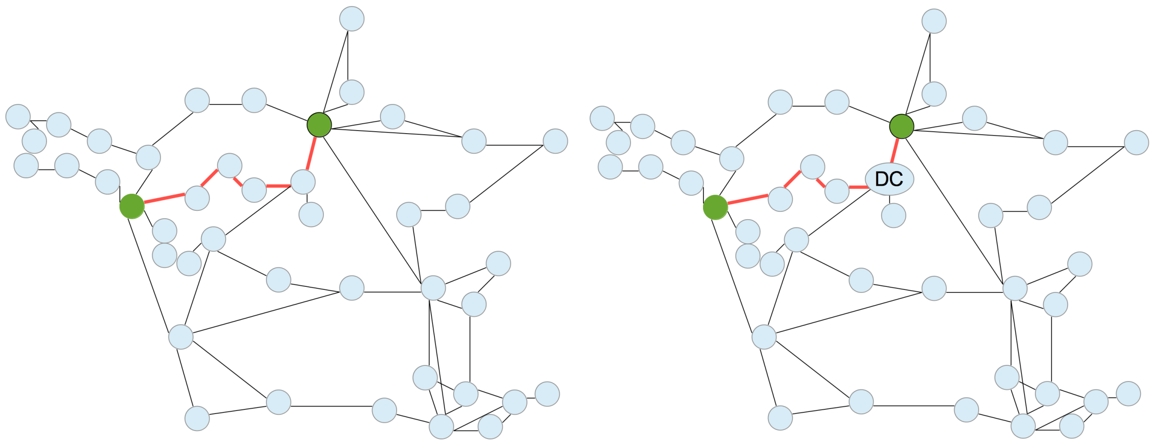
As we stated before, SPO finds the greenest path between A and the central node (C), and B and the same node (C), and DEEPACC finds the greenest path between A and B. If C is part of the greenest path between A and B ($DEEPACC_{Path}(A,B)$), then this node is already part of the set of nodes found by DEEPACC. Thus, $DEEPACC_{Path}(A,B) = SPO_{Path}(A,B)$. Then, the energy consumption of both paths is the same. Formally defined in Equation 4.6.

$$Energy\_Consumption(DEEPACC_{Path}(A,B)) = Energy\_Consumption(SPO_{Path}(A,B)) \tag{4.6}$$

However, if C is not part of the path found by DEEPACC, then $DEEPACC_{Path}(A,B)$ $\neq SPO_{Path}(A,B)$. Thus the path determined by DEEPACC consumes less energy than the one determined by SPO, as described in Equation 4.7.

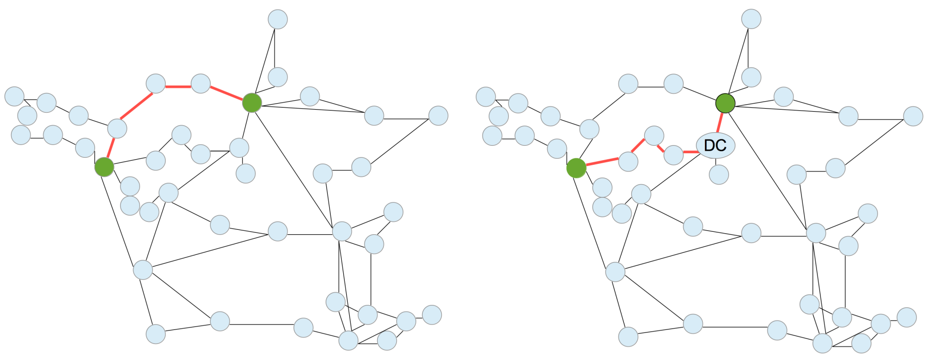
$$Energy\_Consumption(DEEPACC_{Path}(A,B)) \leq Energy\_Consumption(SPO_{Path}(A,B)) \quad (4.7)$$

Therefore, the energy consumption in a microcloud is always smaller or equal to the number of working nodes in an edge Cloud which caches information from a datacenter in the PoP. Figure 4.7 describes the case when the greenest route found by DEEPACC passes through the central node. Figure 4.8 describes the case when DEEPACC finds a greenest route which does not go through the central node.



(a) DEEPACC best route (b) SPO best Route

Figure 4.7: Comparison SPO-DEEPACC when the greenest route passes through the datacenter



(a) DEEPACC best route (b) SPO best Route

Figure 4.8: Comparison SPO-DEEPACC when the greenest route does not pass through the datacenter

In our next experiments, we activated nodes minimizing the number of hops between them. We started by a random node, namely i and activated one of its direct neighbors. In every loop, we randomly choose an activated node and activate one of its direct neighbors. The experiment compares the energy results of both DEEPACC and SPO. Each experiment has been run 10 times to provide a statistically significant result.

Figure 4.9 shows the result of these experiments. Again, it is shown that DEEPACC exhibits always a better energy-efficiency than SPO. The result of these experiments is explained because of the same circumstances as the previous ones. Due to the nature of the approaches, energy consumption of DEEPACC can only be equal or lower than SPO. As expected, the difference between energy consumptions decreases when the number of client nodes increases. That is because, as the number of nodes in DEEPACC increases so does the probability of including the central node. Once the central node is part of the microcloud, the path found by DEEPACC is the same as the one found by SPO.

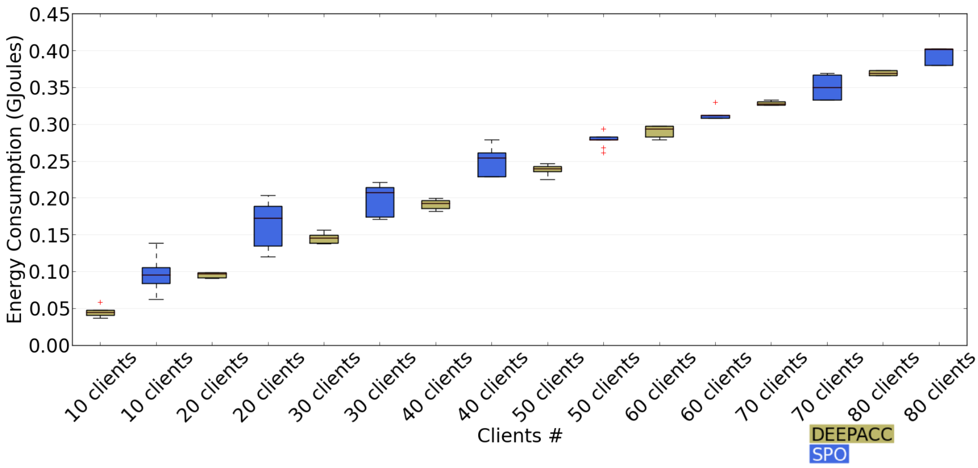


Figure 4.9: Sequential increase of active client nodes

As it has been stated in Chapter 2, energy consumption of a Cloud is divided into three infrastructures: Datacenter, network and client side. As a consequence, once the energy consumption of the network Cloud has been determined, it is necessary to evaluate the impact of GRaNADA on the energy consumption of other actors: datacenter and client side. To do so, it is needed to model the energy consumption in all remaining infrastructures. In Section 4.3.2, the model that we used for energy consumption in datacenters' is presented. Section 4.3.3 addresses the problem of modeling energy consumption in the client side.

4.3.2 Energy consumption of the datacenter part

Works like [196, 274, 275] set the power draw of an average 5,000 sq feet datacenter to 1,127 kW (27,048 kWh per day) with an average PUE (Power Usage Effectiveness) of around 1.8. This configuration is relevant, for the trend in Cloud providers is either to build datacenters of this size in different locations or to divide each datacenter in rooms of about the same size [77, 196]. Thus, we will use this value as the comparison base for datacenters energy consumption.

4.3.3 Energy consumption of the client part

For the energy consumption in the client side, we have obtained experimental values through our own model, due to the lack of a suitable survey on clients' energy consumption in literature. For our experiments, we have used a MacBook Pro - Retina, 15-inch, early 2013 with 8 GB 1600 MHz DDR3 memory and a 2.7 GHz Intel Core i7 4-cored processor. We have estimated the capacity of the battery in Joules, and compared the length in battery life under two different utilization profiles:

- **Average utilization:** Utilization of the computer running average energy consuming applications (browse Internet, music play, etc.);
- **VM simulation:** Similar utilization to the previous one, while running a simulation of GRaNADA over a VM. The VM used runs Debian OS, with 8MB base memory and 16 virtual processors (no graphic interface running), hosting both the service manager and the service provider.

This experiment measures the case of a single private laptop hosting both the service manager and the service provider of the same microcloud. As stated in Chapter 3, both service manager and service provider are hosted in stable and static nodes, so it is not likely that any of these roles is hosted in a smartphone or mobile device with battery restrictions. On average, we found a difference of utilization between the two experiments of 47.3 Watts (67.79 W under average utilization and 115.10 W with the addition of the VM).

4.4 Energy Consumption Estimation of a Traditional Centralized Cloud vs Microclouds

Using the data obtained by our experimentation, we draw a comparison between GRaNADA's energy consumption and the energy consumption in traditional datacenter-centralized Clouds. As discussed in Chapter 2, in a datacenter-centralized Cloud every user consumes, while using an application, a total energy which is divided between the needed energy to provide the service (datacenter side), the energy spent to connect the user to the Cloud service (ISPs' network side) and to make use of the service (client side). It is clear that reducing the computational weight on both datacenters and networks would reduce the energy consumption in the whole Cloud, as long as the client's side energy consumption remains the same or increases only by a factor smaller than the energy saved. This is achieved by GRaNADA through an aware utilization of the networking resources, which gives both the ISP and datacenter provider the ability to downgrade and/or switch OFF and stand-by unused resources. The distribution of energy consumption in GRaNADA is as follows.

1. In the ISP network (*Energy Consumed Externally*), by reducing the distance between client and server, as shown in Section 4.3.1. As explained in this Section, by reducing the number of hops involved in the communication process we are able to reduce the energy consumption in the network. It remains up to the ISP, however, to switch OFF/stand-by or downgrade the configuration of the device. Furthermore, unlike fully distributed approaches, GRaNADA does not replicate data across all nodes. The lack of replication reduces the control traffic in the Metropolitan Area Network (MAN). As it has been shown in Section 4.3.1, an increase in control traffic due to replication implies that nodes in the traffic path need to be upgraded/switched ON, which increases energy consumption.
2. Inside the datacenter (*Energy Consumed Internally*), through the reduction in the inefficient internal communication, as shown in Section 4.3.2. In our system, the

only processes in which the datacenter intervene are the creation and destruction of microclouds.

3. In the client devices, as shown in Section 4.3.3. Energy is increased in the clients' side. However, this increase is only experienced by some clients, that receive the role of service providers. Since computation of a LVM is lighter than the computation of a normal VM, the increase in energy consumption is smaller than its savings, as shown in Section 4.3.3. This result can be extrapolated to networking devices such as domestic routers or ISP networking devices, as shown in Chapter 3. Also, the client side does not need environmental energy (used to refrigerate servers), which is also subtracted from the energy consumed by datacenters.

According to our experiments, the energy saved using GRaNADA compared to a centralized system is of 42 GWatts only in Renater's backbone. From a strictly energy point of view, this implies that the network can host up to 898,340 clients (6,416 clients per node in RENATER), all of them hosting the LVM before the energy saved solely by reducing the path in the network becomes less than the energy spent by hosting the computation. Also, using the figures described above for Cloud datacenters (1,127 kW), we extrapolate that the client side can run the equivalent of 23,978 LVMs with the same energy consumption. It is equivalent to running 8 VMs per server (assuming a central datacenter of 5,000 sq feet with 100 racks and 30 servers per rack) or creating 23,978 microclouds - more if the LVM is split. Summing up the network savings, we reach 922,167 microclouds (6,587 clients per node, assuming them all play the providers' role) before the energy expenses overcome the energy savings.

Even if there exist variations in energy consumption in the nodes, GRaNADA shows large savings in energy consumption compared with the predominant datacenter-centralized architecture. It also shows an important improvement in energy consumed with the other distributed approaches such as edge Clouds.

4.5 Conclusions

In this chapter, we evaluated the energy issues of Cloud networks. To do so, we modeled the energy consumption of a Cloud application, and we provided a simple yet complete model of energy consumption in the Cloud.

Then, we showed how GRaNADA can be instantiated in a core network, presenting an example on a real-life core network (RENATER). The goal of this chapter is to evaluate the energy improvements of using a distributed approach. To do so, we used the aforementioned adaptation of GRaNADA and simulated its performance using a packet-level simulator. This tool gave us a very thorough vision of the network level processes run. We compare the results of our approach to two main approaches shown in Chapter 2: datacenter-centralized and edge Cloud. Experiments show that bringing the computation closer to the user reduces the amount of energy consumed. They also show that, by reducing the replication of the VM, more energy can be saved. Our simulations exhibit a large decrease in energy consumption in the network when the computational power (VM) is distributed. Using a distributed approach where the VM is cached in the edge, as analyzed in Chapter 2, showed a significant reduction on energy consumption. This energy consumption could be reduced to a higher degree when the VM is hosted in the edge with no replication (GRaNADA's approach). Finally, we showed that by distributing the computation without replicating data, energy is also saved in the datacenter, where the computation is decreased, and increased in the clients side. However, the energy consumed in the client's side remains smaller than the energy saved both in the network and in the datacenter.

Results of the experimentation are encouraging, but they leave aside some questions. Our system is designed to reach its best in heterogeneous networks (different networking devices), where the routing path and provider's selection are more important. Next steps in experimentation will be studying a more localized network infrastructure with different applications, dynamic configurations and heterogeneous users. Once compared the energy consumption of our approach with state-of-the-art approaches, the Quality of Service experienced by the user has to be considered.

Chapter 5

Microcities: GRaNADA for Smart Cities Applications

In this chapter, we propose an integration of GRaNADA in a smart city infrastructure to provide a platform for mobile Cloud computing. To do so, GRaNADA creates local micro-clouds merging static public devices, such as the smart city infrastructure and networking equipment belonging to the ISP, and private static and mobile devices (i.e. computers and the users' mobile devices). We consider these devices to be located across a given bounded geographical area, typically a neighborhood in a city. GRaNADA provides the smart city infrastructure with lightweight mechanisms to handle the dynamism of a mobile Cloud. Users may arrive at or leave the considered geographical area, as well as move inside the boundaries of the neighborhood. Also, GRaNADA eliminates the need for dedicated infrastructures (i.e. datacenters) and provides a dynamic and tailored environment where multiple services coexist.

The remainder of the chapter is as follows: Section 5.2 describes the scenario on which we will test our approach. Section 5.3, summarizes GRaNADA's design shown in Chapter 3 and how this design is mapped into the case study infrastructure. In Section 5.4, we present our experimentation methodology and describe the results obtained. Finally, Section 5.5 highlights the lessons learned from this chapter.

5.1 Motivation

In Chapter 2, we described how Clouds can provide a support infrastructure for mobile devices to offload computation. Offloading computation in the Cloud allows mobile devices to use applications which are, otherwise, restricted to them because these applications demand more resources than what these mobile devices can provide. Indeed, one of the main requirements set by mobile devices is high-speed connectivity between the clients and the Cloud. As explained in Chapter 2, low latency is a requirement for real-time applications to provide good Quality-of-Service to the users.

As studied in Chapter 2, a significant portion of traffic in the Cloud is localized inside a bounded area, especially in mobile Clouds. While the proportion of geographically constrained traffic is difficult to estimate, similar data to our scenario of traffic characterization in different areas can be found in literature. For example, in [276] the authors evaluate the consumed and generated traffic in a rural African village. In this scenario, the authors show that most of the generated and consumed traffic is of a local scope, with web and social networking the most utilized services. Also, in [277], the authors characterize usage of a freely available outdoor wireless network in California (USA). Their results show a peak of smartphone connections in transit areas, while in residential areas the connections are more balanced between static and mobile. Also, in commercial areas they show a higher activity in both.

We claim that the datacenter-centralized architectures are not suitable for offloading mobile computation in a mobile Cloud. As described in Chapter 2, in datacenter-centralized architectures, data are sent to a central infrastructure (i.e. datacenter) which processes data. This distance leads to increased utilization of the broadband Wide Area Network (WAN) and poor user experience. This situation is aggravated when the datacenter is located in a different region. For example, using Amazon's Cloud infrastructure [38], European users have the possibility of using two different datacenters. The first one is located in Ireland, while the second is located in Germany. As the distance between the user and the datacenter increases so does the latency, as studied in Chapter 2.

Indeed, Fernando et al. [35] argue that considering access fees, latency and bandwidth of wireless connectivity, a Mobile local Cloud - constrained to the location of the user - is a better alternative than a remote one. Furthermore, local Clouds also provide context awareness [149], which is another requisite of mobile Clouds [35]. We believe that a distributed Cloud, such as GRaNADA, can provide a better Quality of Service (QoS) for Mobile Cloud networks large urban populations, by confining local traffic close to the user while still maintaining the advantages of centralized scenarios defined in Chapter 2.

The aim of this chapter is using GRaNADA to improve the Quality-of-Service delivered to the users of mobile Clouds. We give special relevance to reducing latency as part of the QoS and propose using GRaNADA in a realistic geographically bounded scenario, described in Section 5.2. In this scenario, static and mobile private devices are integrated inside a public infrastructure, i.e. a smart-city infrastructure. In this case study, each area is bounded in a neighborhood, where users share certain interests.

5.2 Case Study: Neighborhood Services

Applications targeting the specific population of a neighborhood are a good example of geographically localized services. From a platform perspective (i.e. the deployment of neighborhood applications), many services are only of interest to the population of a community. For instance, information about street works, water or electricity cuts or local store information, such as goods in stock or opening hours, affect only neighbors of the area, who benefit from these utilities. Indeed, this type of applications have already been assessed by software applications such as NextDoor [278] and GoNeighbour [279]. These services are social networks where users share information of interest only with their neighbors. In such applications, data are usually created and distributed in the same region. Furthermore, involved actors participate as creators and consumers of data at the same time. For example, in the case of NextDoor [278], information is created and shared in a social network by users who live in the same neighborhood.

On the other hand, we consider the case study from an infrastructure perspective. In urban environments, some public infrastructures are used to control services, such as waste management [280] or provide information about the state of the roads inside the city [281]. These infrastructures are part of the Smart Cities initiative. The smart cities initiative [282, 283], aims at creating a city-wide infrastructure of devices which provides information about areas of this city and services to its citizenry. Smart cities' adoption is booming [284], and it has been already adopted by several cities around the world such as Barcelona [285], New York City [286] and Beijing [287]. As studied by Delmastro et al. [288], users' interaction with smart cities is improving the participation of users as providers of digital services. Indeed, according to the International Electrotechnical Commission, users are increasingly becoming providers of services and not only consumers thanks to smart cities [283]. This is explained because smart cities provide an infrastructure which, at the same time, provides and consumes information. For example, smart cities can be used to provide information about transportation to the users [281]. In this example, data are collected from users around an area, such as a metro station, and served to users interested, such as users traveling to the same station.

In this chapter we propose creating a Cloud where users and smart cities infrastructures can collaborate, leveraging from the already existing infrastructure. In Section 5.3, we show how GRaNADA can be used for providing a good Quality-of-Service to users of such a Cloud.

5.3 GRaNADA for Smart Cities: Microcities

In the case study described in the previous section, each instance of a service exists for, and only for, the population of the neighborhood. Also, in this case study, an infrastructure has been deployed by the city council to provide computational services.

As studied in Chapter 2, distributed approaches provide a better Quality-of-Service, especially in term of low latency, than centralized ones when users of the service are geographically close to each other [35]. GRaNADA has been designed to exploit such cases where the users of an instance are located in a bounded area (see Section 3). GRaNADA is also designed to merge static and mobile devices transparently to the users in microclouds and can leverage from the existing static infrastructure (i.e. the smart city infrastructure) to ensure a better Quality-of-Service to the user.

We propose integrating GRaNADA into the smart cities infrastructure to provide a platform for supporting mobile Cloud applications. We call this platform *microcities*.

5.3.1 Microcities' design

We consider three actors in the microcities' Cloud: the smart city infrastructure, the ISP infrastructure and the client side infrastructure. Both the smart city and the ISP infrastructure are public infrastructures, while the clients' side infrastructure is formed by private devices. Private devices can be either mobile, such as smartphones or tablets, or static, such as computers or domestic routers. On the other hand, public devices are always considered to be static. Figure 5.1 represents a map of a real-life neighborhood hosting a smart city infrastructure (example in Barcelona). In this figure, we depict a schema of a possible deployment of GRaNADA. Actors are represented using different colors, and a visual representation of possible active microclouds in the area (such as discussed in Chapter 3) is provided.

As it has been described in Chapter 3, GRaNADA assigns different roles to each device. Each role carries a functionality, such as management or LVM hosting. Some other roles have no specific functionality associated with them, and are given the client role. As a reminder from Chapter 3, the available roles in GRaNADA are the following.

Base Infrastructure (BI), which is used as a backup and for centralized management (if necessary). It can also be used to run the LVM if no other node is available.

Super Manager, which controls several microclouds. It can determine when one or more microclouds are being inefficient and may provide a better configuration by splitting

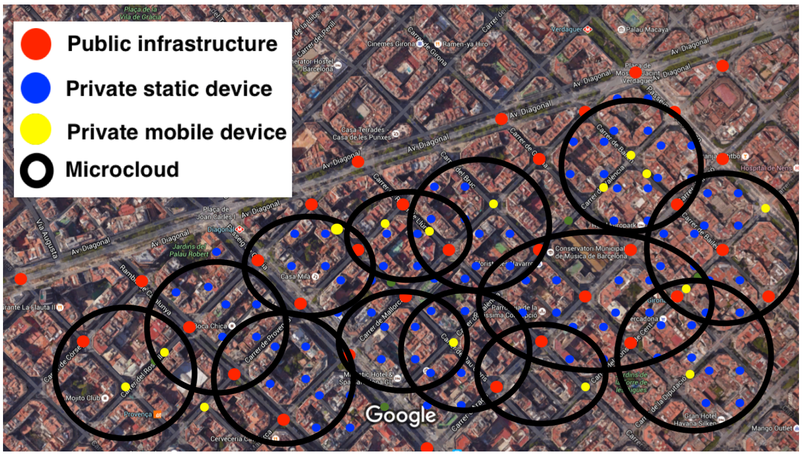


Figure 5.1: Microclouds operating within a neighborhood (image: Google Maps)

or merging microclouds. It also decides which device is assigned the service manager role inside a microcloud.

Service Manager (SM), which is in control of the internal management inside a microcloud. It decides which device is assigned the service provider role and determines an efficient topology for the microcloud. This role is unique in the microcloud.

Service Provider (SP), which is the device hosting the Light Virtual Machine. This role is unique in the microcloud.

Client, which is given to any device consuming or contributing information.

Roles are assigned according to the resilience of the device. For example, GRaNADA assigns the management roles to static and resilient nodes. On the other hand, mobile nodes are seldom assigned any other role than client. Indeed, any static node with enough available resources has a priority to be assigned a management or provider role. Figure 5.2 shows a schema of all roles participating of a microcloud. In this figure, microcloud A and B represent different services, overlapping on the same network and, sometimes, on the same devices. In Sections 5.3.2 and 5.3.3, we describe how roles are assigned in the infrastructure.

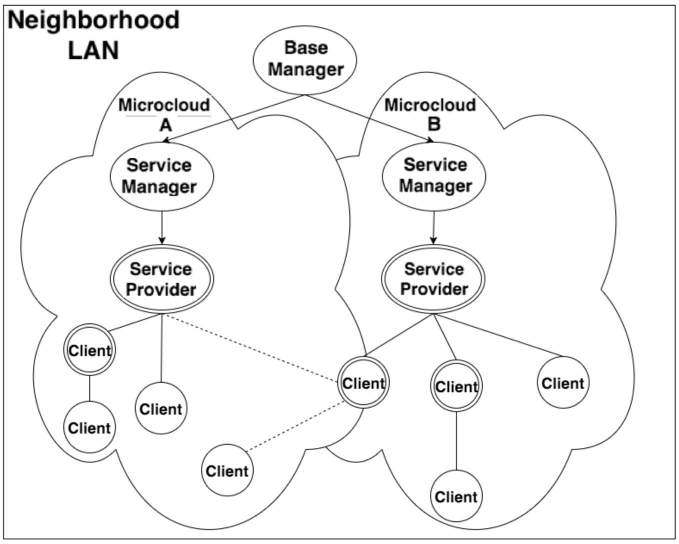


Figure 5.2: Logical distribution of microclouds

5.3.2 Public infrastructure

The public infrastructure is composed by all devices managed either by a company or the city council. Public infrastructures are especially relevant for the base infrastructure and super manager, given that it is formed by trustworthy, stable and resilient nodes. However, devices in these infrastructures can take any other role, such as client role (when the device provides information), service manager and service provider. In our work, we consider the following public devices.

ISP network infrastructure, which is managed by a company [289]. The devices managed by the ISP are network devices, such as routers and switches, and servers. They are all located in a Point-of-Presence [267], as explained in Chapter 4.

Smart city infrastructure, which is either managed by a company [290] or by the public council [291]. The deployment of a smart city infrastructure is distributed within the city, with devices in different locations [283]. This category includes different devices such as wireless access networks or Cisco's mesh outdoor network [292].

Public multipurpose infrastructures, which is managed by private companies. In this category, we include devices offering support to services provided by companies, such as coffee shops or retail stores. For example, network devices used to provide Internet access to clients of a store [293].

5.3.3 Private infrastructure

The private infrastructure is composed of all devices which are part of a private network for one or more users. The considered devices are the following.

Computers/Laptops, which are expected to be the most common client devices, along with mobile devices. The difference with mobile devices is that personal computers are static and have a longer lasting battery, thus they are assigned both the service manager and the service provider roles with priority over mobile devices.

Mobile devices are different from the previous category, as they only take the client role.

Home routers, which can be used for hosting either the service manager or the service provider roles to extend the capacity of the network. Many home routers are nowadays manufactured with an application layer [294], which is a main requisite to host the service provider role, as explained in Chapter 3. The client role won't be given to home routers, as they don't consume or provide information.

Home servers, which can take a service manager or service provider role. They can also take the client role, as they can provide information.

Table 5.1 shows a summary of all possible roles and which devices can host them. For each role, it is assigned to the device with available resources with a higher priority. In Table 5.1, devices are ordered from high priority to low priority (top to bottom).

5.4 Evaluation

The aim of our experiments is to evaluate the suitability of GRaNADA operating within a neighborhood's mobile Cloud from the point of view of the Quality-of-Service experienced by the users. To evaluate the QoS, we focus on the average packet transmission delay between nodes, the overhead of the routing protocol (extra communication time), packet loss probability and the reliability of the system. We focus on these variables because of how they are affected in GRaNADA. First, as explained in Chapter 2, the average latency experienced by the user is improved when the distance between the client and the VM is

	Public infrastructure	**Private infrastructure**
Base Infrastructure Role	All	None
Super Manager Role	All	None
Client role	All	Home Servers Computers/Laptops Mobile devices
Service Provider	All	Home Servers Home routers Computers/Laptops
Service Manager	All	Home Servers Home routers Computers/Laptops

Table 5.1: Roles assignment in the system

reduced in distributed architectures. Thus, we evaluate how GRaNADA can decrease the average packet transmission delay between nodes. Second, as described in Chapter 3, a microcloud's topology changes dynamically either during the creation of the microcloud or changes in the connections between nodes. When the microcloud's topology is changed, GRaNADA needs to compute the new connections between nodes and relocate the Light Virtual Machine (LVM). This process affects the user, as it requires extra time. Measuring the overhead of the routing protocol, we evaluate how the dynamism of the mobile Cloud affects the users' experience in GRaNADA. Also, the dynamism of the system affects the probability of packet loss. For example, during the migration of the LVM, packets are lost. We measure how the dynamism of the mobile Cloud affects the coherence of data experienced by the user by measuring the probability of GRaNADA losing packets. Finally, the reliability of the system is measured by estimating the probability of the users losing stored data due to failures in the infrastructure (for example, a Service Provider disconnecting from the infrastructure).

Similarly to the experiments shown in Chapter 4, we compare GRaNADA to a datacenter-centralized Cloud. Again, we acknowledge the existence of other approaches not contemplated in the evaluation, such as P2P Cloud computing. However, they are already represented by one of the solutions. For example, the case of P2P Clouds is represented by GRaNADA from a QoS point of view. On the other hand, the edge Cloud case is already covered by the centralized approach. While distance between users and the VMs in edge Clouds is reduced compared to the centralized approach, latency is fixed in both cases as VM are hosted in a datacenter. Thus, the edge Cloud represents, to our experiments in the current chapter, a centralized approach where the datacenter is located near the user.

Using a real-life infrastructure in a smart city [282] would require a set of computing resources which is beyond our capacity. As an example, in a smart city such as Barcelona every neighborhood has between 542 and 56,503 people, as stated by the latest census [295]. Assuming one device per person, and adding the necessary network and smart city infrastructure, the resulting average amount of devices per neighborhood would accounts for more than 25,000 devices.

In consequence, we have used a synthetic neighborhood infrastructure in our evaluation. However, in order to prove the viability of GRaNADA we also used a real-life prototype.

We divided our experimentation into two parts. In the first part, we present the results obtained running a prototype in a small number of nodes. We also compare these results to a centralized architecture. Later, we extrapolate these results into a simulation of a neighborhood's network and compare the 2 architectures (centralized and GRaNADA).

5.4.1 Prototype

In order to show the viability of using GRaNADA in a real-life environment, we deployed a prototype of GRaNADA using 10 nodes in a Local Area Network (LAN). These experiments have been used to obtain real-life data about latency and packet loss probability. GRaNADA's infrastructure was deployed as follows. We used 10 nodes, 6 laptops (4 MacBook Pro 2.7 GHz Intel Core i7 8 GB 1600 MHz DDR3, 2 HP EliteBook 2.10GHz Intel Core i7-4600U CPU 16GB 1600 MHz DDR3), 1 multipurpose small computer (Raspberry Pi 2 model B 900MHz quad-core ARM Cortex-A7 CPU 1GB RAM), 2 smartphones (OnePlus One Qualcomm Snapdrago 801 processor with 2.5GHz Quad-core CPUs running CyanogenMod 11S based on Android 4.4 and iPhone6 Dual-core 1.4 GHz Typhoon ARM v8-based running iOS8) and 1 network switch to which all are connected (DELL PowerConnect 6224). Connections used where WiFi for the case of smartphones and Ethernet in the case of laptops and multipurpose small computer. Among the computers, one has been chosen as a service provider, and the rest as clients.

On the other hand, we run similar experiments placing the VM in a datacenter-centralized infrastructure. The centralized infrastructure uses the same infrastructure. The switch is later connected to the Internet, through which the data are sent to and from a mainstream IaaS Cloud, Amazon EC2. We launched experiments in different availability zones and pick the availability zone which provided the smallest Round-Trip Time (RTT) values, due to fairness in the comparison.

The communication process simulates the interaction between client and sever in an on-line shared document application. As explained in Chapter 4, given the lack of traces in literature for concurrent access to multiple users' Cloud applications, we have obtained several real 45-minutes traces from actual Google Drive sessions, using the network packet analyzer tool Wireshark [273]. This traces are obtained from the concurrent use of documents (addition/deletion of text).

As a first evaluation, we compare the communication delay perceived by the users, which is depicted in Figure 5.3. As expected, GRaNADA provides a latency several times smaller than using a datacenter-centralized solution in very localized environments. Indeed, GRaNADA provides an average delay of about 15 ms between clients, while the centralized experiment shows an average RTT of about 117 ms. Results are explained by the distance between users in the case study (almost negligible compared to Amazon's world-wide area).

Second, we randomly changed the location of the service provider in GRaNADA. To do so, we deploy the server software in a new client. Once it is deployed, the former service provider (the node previously running the server software) sends a migration message to all the nodes with the new IP. Finally, all nodes start the communication process with the new service provider. This process was done 10 times. We did not register packet loss during these experiments. This is explained because the protocol used in the connection is TCP, which ensures the arrival of the packets at destination. Also, all clients are aware of the change once it is available.

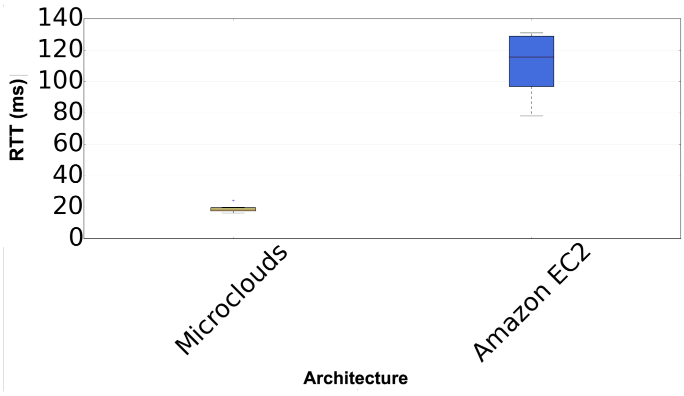


Figure 5.3: Microclouds vs Amazon EC2 RTT

5.4.2 Simulation

We extrapolate the data obtained in the previous section to a larger network. To do so, we simulated a synthetic physical neighborhood topology using NS3 [63]. This simulation runs GRaNADA's software to distribute computation among the nodes. To provide an accurate simulation, NS3 reproduces the whole communication process up to a packet level. A packet-level simulator is needed to obtain results as precise as possible. The results obtained from the prototype have been fed in the simulator to evaluate GRaNADA's expected Quality-of-Service in a larger scale. In these experiments we run a simulation of an on-line data sharing application, using the same traces as before.

Given that NS3 is a packet level simulator, the simulation has been scaled down to interpolate relevant information in a reasonable amount of time, as the computation time of the network simulator is exponential [269]. (For example, simulating more than 100 mobile nodes moving over the static infrastructure takes almost 24h to represent 1 hour worth of users interaction). The simulated network contains a variable number of mobile nodes (between 2 and 100), with a random mobility over a physical network of 45 static nodes, which represents the smart city infrastructure. The infrastructure of the static network is represented in figure 5.4.

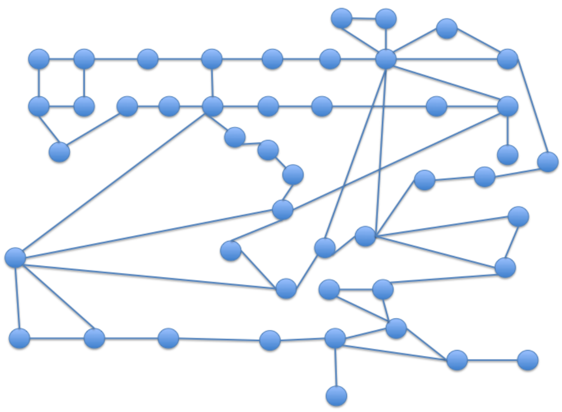


Figure 5.4: Physical topology of the static network

In a first simulation, we evaluate how the amount of mobile nodes affects latency in GRaNADA. Figure 5.5 shows the RTT evolution when the number of mobile nodes in the microcloud increases. As expected, the latency increases with the number of mobile

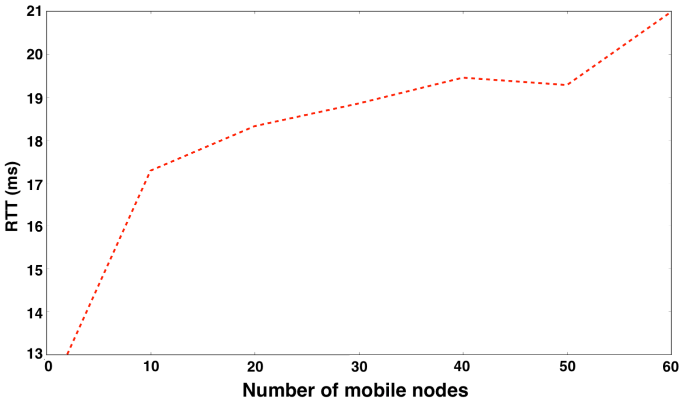
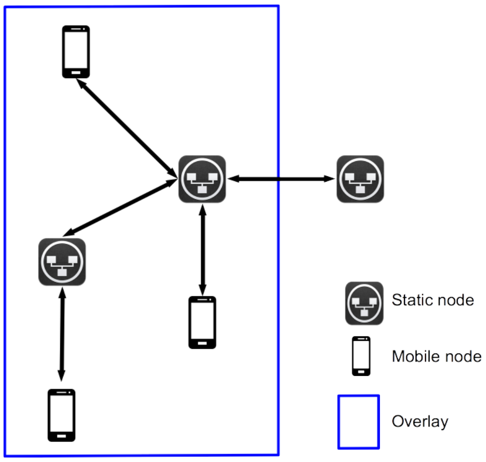
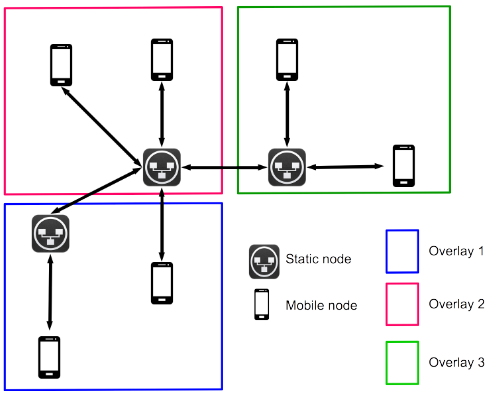


Figure 5.5: Total microclouds RTT in ms

clients and the distance between them. These results highlight that large microclouds (those with a large number of mobile nodes) offer a worse QoS to users in terms of latency than microclouds with a small number of mobile nodes. For example, the difference in average latency between a microcloud with 10 mobile nodes and another one with 60 mobile nodes is about 4 ms. While this difference is not significant, it varies latency between users of the same service when the service is popular, thus inducing heterogeneity among the users' connections.



(a) Schema of 1 microcloud

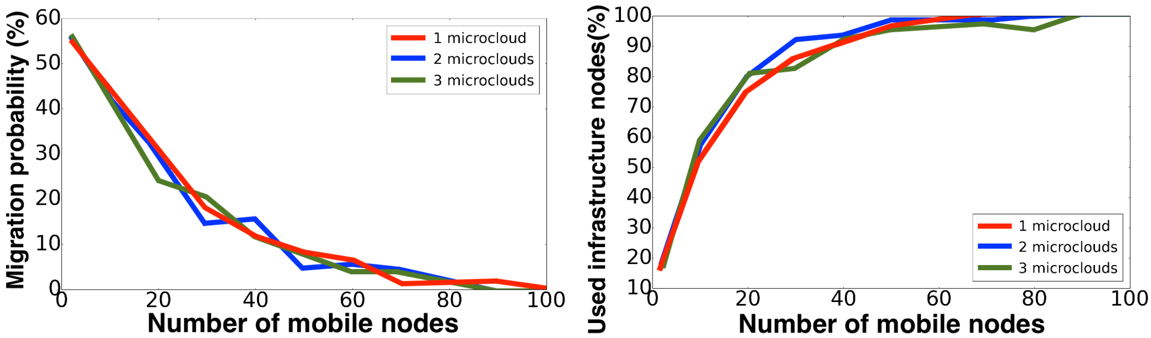


(b) Schema of 3 microclouds

Figure 5.6: Schema of 1 and 3 microclouds

Second, we evaluate the impact that reconfigurations have on LVM migrations and QoS. The mobile nodes' mobility causes reconfigurations, because when a mobile node connects to a new static node, the resulting latency may be very high. A reconfiguration may cause a migration of LVM, if under the new configuration the placement of the LVM is inefficient. As we described earlier, each time that GRaNADA migrates a LVM users stop sending data to a node (former service provider) and communicate with another (new service provider). While these migrations require very small downtime [296], frequent migrations are not desirable. Frequent migrations require computational time and resources. On the one hand, the node acting as a service manager has to contribute resources for calculating the migration of the LVM. On the other hand, frequent migrations may decrease the QoS of the clients because of an increased latency caused by these migrations.

These experiments have been run using different configurations. First, it has been considered that all mobile nodes are part of the same microcloud. This means that an overlay network is created connecting all mobile nodes in the infrastructure, as depicted in Figure 5.6a. After, we run the same experiments with 2 and 3 microclouds. This means that every mobile node in the system is assigned a microcloud. In this configuration there are as many overlays as there are microclouds, each one connecting different mobile nodes, as shown in Figure 5.6b.



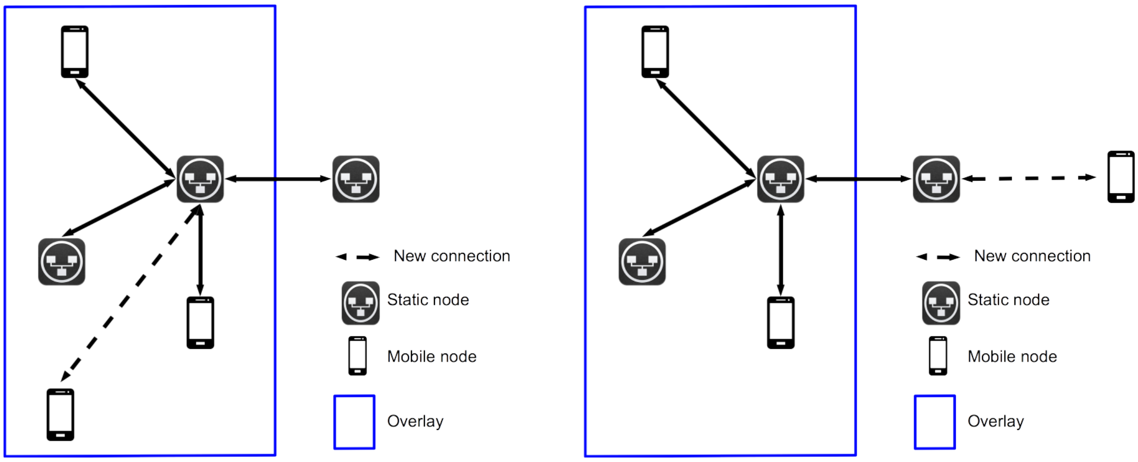
(a) Probability of migration of LVM

(b) Infrastructural nodes used

Figure 5.7: Probability of migration of a LVM vs. utilization of the network

Figure 5.7a shows the probability of migrating a LVM using either one large or several small microclouds' configurations. As a comparison, Figure 5.7b shows the number of devices occupied in the network: a node is occupied if it is part of, at least one microcloud. It can be observed that when the number of occupied nodes increases, the number of migration decreases. This is explained by how DEEPACC behaves in the case of reconnection. As explained in Chapter 3, when a mobile node connects to a microcloud DEEPACC finds the shortest path between the mobile node and any node in the microcloud. Thus, in the case of a reconnection, a mobile node disconnects from a static node and connects to a new one. In this situation, if this new node was already part of the former microcloud, it is possible that other mobile nodes are already connected to it, as it is depicted in Figure 5.8a. Then, latency to the mobile node is the same as the latency to its neighbors, which was considered acceptable. However, if the static node to which the mobile one connects does not participate of the microcloud before, then the added latency may be much higher, since more devices may add more latency, as it is depicted in Figure 5.8b.

Figure 5.7 also depicts the result of running these experiments using 2 and 3 microclouds. In these experiments, we assume that the original microcloud can be split in several smaller microclouds, as explained in Chapter 3. For example, when in the previous experiments we used 10 clients connected inside the same microcloud, in this set of experiments we reproduce the process using 2 microclouds with 5 clients each (or 3 microclouds, with 4 and 3 clients). While each microcloud has fewer clients than in the previous set of experiments, it does not result in a large difference in the number of migrations when deploying several small microclouds. This is because, while the number of clients per microcloud is smaller, the number of occupied nodes in the infrastructure remains similar, as depicted in Figure 5.7b. This means that, while each microcloud has fewer clients, the size of the microcloud is similar to the case when all the clients are merged in one microcloud. This situation is caused by the mobility of the nodes. If a node is far from the others, the microcloud includes several nodes used to connect this node to the microcloud, as depicted in Figure 5.9.



(a) Mobility of a mobile node inside the microcloud (b) Mobility of a mobile node outside the microcloud

Figure 5.8: Mobility of a mobile nodes inside/outside the microcloud

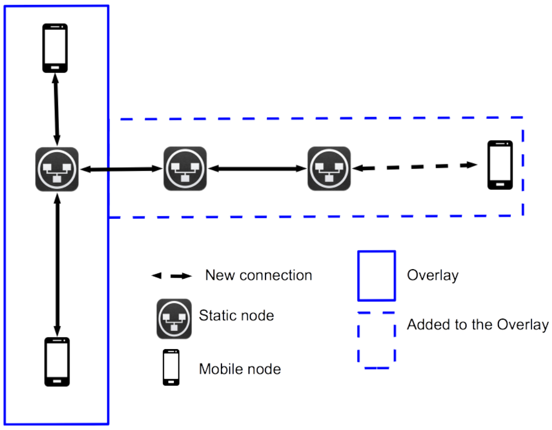
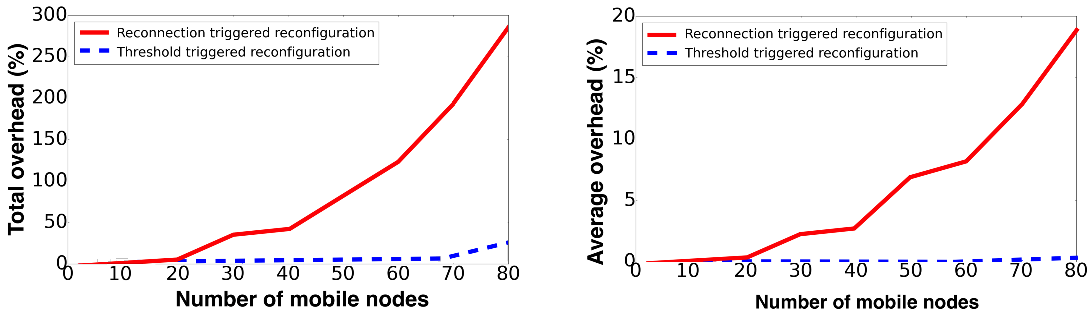


Figure 5.9: Reconfiguration caused by mobility of the nodes

From these experiments, we draw the following conclusions. As shown in Figure 5.7, for small microclouds (10 or fewer mobile nodes), the probability of the LVM being migrated is around 50%. It rapidly decays with the addition of new nodes as the microcloud's topology converges to that of the physical one until all the nodes participate in the overlay network and so no migration is needed. This situation shows the need for an heuristic discussed in Chapter 3 to reduce the number of migrations whilst providing a good QoS. While migration in datacenter-centralized approaches are not common, in our approach these are done to adapt to changes in the topology, which may be frequent considering mobile nodes. Thus, when the network is populated enough and with the utilization of a suitable threshold which assures there is not an excessive number of migrations, this characteristic becomes an advantage. This latency threshold depends on both the application and the requirements of the users. Thus, it can be either set statically by the service administrator (one maximum latency for all microclouds) or dynamically through negotiation between users. The case where users negotiate their threshold is studied in Chapter 6.

In our third set of experiments, we measure the overhead computation of GRaNADA. The increase in the network's size is followed by a higher overhead in the computation time needed to map the overlay network (using the real one in between). Furthermore, as stated before, migrations of LVMs produce extra computation. While the service is not halted until a new path is computed and established, excessive extra computation affects the QoS and the demands the platform makes of the finite resources. An inefficiency threshold can be also used to avoid the excessive computational overhead caused by an excessive number of reconfigurations (even if it does not lead to migration). This inefficiency threshold ensures that a reconfiguration is only started after when the system becomes too inefficient. We used a threshold where no reconfiguration is launched until, at least, 50% of the mobile nodes involved in the original overlay network change their location. While in our experiments mobile nodes move randomly, 50% has shown to be a fair threshold.



(a) Total overhead on the join process

(b) Average overhead on the join process

Figure 5.10: Overhead (in % of time) compared with the ideal case (no overhead: 0%)

Figure 5.10 compares the computational overhead of a threshold-triggered redistribution (using our heuristic) and a rejoin triggered redistribution (that is, launched when a client joins or rejoins the network). In the second case, the random mobility of the clients cause, every time that one client detaches from one node and connects through a different node, all clients participating in the network to start the join process; new routes are calculated and the LVM is migrated to a new node if the latency is higher than the threshold. If there is no threshold then the location of the LVM is computed every time that there exists a reconfiguration. Computational time is compared to the ideal situation where no management is needed (because there is no mobility of nodes, thus no rejoin process is needed). As shown on Figure 5.10a, the total computation time added by the join process exceeds almost 3 times the time used by the application to interact with clients. On the other hand, by using a threshold triggered approach, the management time remains under 25% of the interaction time. On average - comparing to the total execution time -, as shown in Figure 5.10b, the use of a threshold triggered approach is almost imperceptible compared to the rest of the computation time. As before, the utilization of a good threshold is essential to ensure that there are no excessive reconfigurations.

Finally, we evaluate the performance of the manager in our system. To do so, we compare the extra computational time added to the service manager for the configuration of the microcloud. This overhead has been measured using a MacBook Pro, with a 2.7 GHz Intel Core i7 and 8 GB 1600 MHz DDR3, as a service manager. It is expected that the size of the microcloud affects the computational time, as the service manager has to plan longer routes. Thus, our experimentation compares the performance of different approaches: one microcloud managing all clients, or the option of splitting this into several autonomous, and sometimes overlapping, microclouds. Following the design specifications, one service manager controls only one microcloud, thus several microclouds are managed by as many

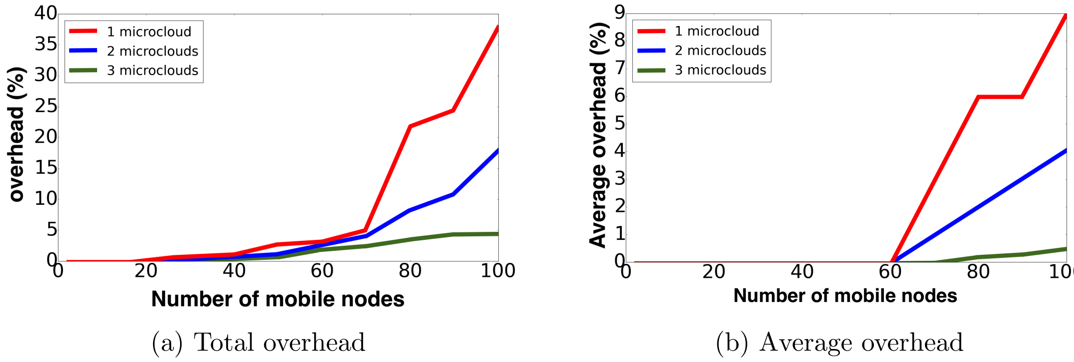


Figure 5.11: Performance comparison of the SM splitting microclouds (in % of time)

independent service managers. Figure 5.11 shows the total and average overhead along the duration of the trace.

As we can see in Figure 5.11, the overhead increases exponentially when the number of clients increases. This is because having more clients not only affects the size of the network, which makes the service manager compute longer routes, but also the number of rejoins. That is, the more mobile nodes in the system the higher the probability of a node changing location, thus triggering a reconfiguration. As shown, when the network size is around 100 clients, the total overhead reaches a 40% of the time. However, Figure 5.11a shows that, by splitting microclouds, this time can be significantly reduced. Since the path computation is distributed among different nodes, the average overhead computation on each service manager is reduced. Given that each microcloud manages fewer nodes, the total overhead is also reduced.

5.5 Conclusions

With the emergence of personal mobile devices, a large amount of data is being generated and consumed everyday. This amount of information is expected to keep growing as the technology gets more mature and the number of users adopting it increases [11]. This situation is especially acute in the case of geographically constrained information. In many cases, the geographical distance between clients is very small compared to the distance between the clients and the data processing and storage servers of centralized Clouds. Whilst, by design, this situation is supported by Cloud computing, the existing implementations, employing large centralized datacenters, become a bottleneck when it comes to latency, network flooding and the provision of resources.

In this chapter, we evaluated the suitability of GRaNADA, a distributed approach for managing services used in a limited geographical area - such as neighborhoods - in the context of mobile Clouds. As it has been shown in Chapter 3, GRaNADA is designed to be distributed and network-oriented, that is, it considers the network as a participating entity in the Cloud and takes advantage of it. Thus, GRaNADA exploits network resources to reduce unnecessary data transportation over long distance networks, running computation on the nodes participating in the communication, like personal devices, network equipment and/or specific purpose hardware such as smart cities networks. We show the benefits of using our architecture over the currently dominant datacenter-centralized approach in terms of QoS. Finally, we show that, while our system requires extra management computation from participating nodes, this overhead is significantly reduced due to its adaptability and the use of a threshold-based heuristic. Additionally, this extra management, needed for route planning and LVM migrations, allows the system to provide better QoS than in the centralized case.

One main concern for the adoption of our system is the unfair distribution of load, where some devices run computation while others are free riders. The next chapter proposes an incentive system for adopting a microcloud approach, targeting both the free-riding problem and the definition of a Quality-of-Service threshold based on the users' demands.

Chapter 6

Micromarkets: Incentives in the Use of Microclouds

In this chapter, we extend the microcities platform presented in Chapter 5. We present an incentive model for a mobile Cloud based on auctions. We apply this model in the microcities platform to encourage users to share their resources in a profit-driven market. On the one hand, users willing to share their resources compete to offer the best service in exchange for a compensation. On the other hand, users who want to offload computation compete to locate the LVM in a device which provides them a good QoS. Finally, we allow the application provider supervising the process to participate, so a minimum QoS is guaranteed to all users.

The remainder of this chapter is as follows: Section 6.1 presents the problem tackled and the scenario. Section 6.2, presents our solution. An evaluation is shown in Section 6.3. Finally, Section 6.4 highlights the lessons learned from this chapter.

6.1 Motivation

In former chapters, we have shown the need for distributed clouds to handle the ever-growing demand of computation. We have provided a distributed architecture which allows users hosting different services. In this context, Quality-of-service and SLA guarantees are considered in literature as one of the major challenges of cloud-based services [297]. However, in distributed architectures where computation is hosted in users' devices, respecting the SLA depends, in many cases, on the will of the devices' owner. Distributed clouds may be unfair when some users contribute more resources than others. This case is especially acute when a significant percentage of the users do not contribute at all. These users, called free riders, are estimated to account for the largest percentage of users in collaborative networks [139]. The unfairness associated with the free riding problem represents an obstacle to the adoption of a collaborative approach, such as GRaNADA.

To address this problem we propose a lease-based mechanism. A *l*ease is a contractual arrangement between an entity, which rents part of its computational power, and a group of users, which offers a payment in return. In a lease the entities offering the computational power are called *s*ellers. The sellers rent part of their resources to host the Virtual Machines (VMs) used to provide the service. On the other hand, the rest of users, called *b*uyers, pay the sellers for hosting the VM.

However, in the case of distributed systems, this lease is commonly assumed to be based on the good-will of users, i.e. it assumes that buyers are willing to share their resources without any compensation. As shown in [139], this is far from being realistic. We consider the concept of leasing in distributed systems needs to be adapted to the dynamism of mobile Cloud computing. In our leasing, all users able to host computation become at the same time sellers and buyers. On the one hand, users act as sellers, offering their resources to host computation in exchange for a compensation. On the other, they also act as buyers, compensating other sellers for the use of their resources. As a consequence to this new concept of lease, a pricing system is required.

6.1.1 Objectives

In this Chapter, we propose a multi-sided auction system, where the user becomes both buyer and seller, auctioning on the resources of other users as needed. Furthermore, we propose an open auction system where the application provider supervises the process, and has the possibility of bidding along with one or more users if the result of the auction is unfair to some users.

6.1.2 Scenario

We have used the same neighborhood scenario as defined in Chapter 5. The characteristics of this scenario that are of interest to this chapter are:

Highly dynamic: The mobility of the different mobile nodes produces a highly dynamic system. Thus, auctions have to adapt to this dynamic nature.

Client orientation: The system needs to take into account different users requirements (such as CPUs, RAM, storage space or expected QoS). Every client may request different conditions for the SLAs.

Minimum satisfaction: End-users' QoS is a main concern in the design of the system, as interactive applications are considered. Thus, a minimum satisfaction needs to be guaranteed to nodes.

Credit mobility: Users' credit can be used to pay for different applications in different microclouds. If users move between neighborhoods, their credit has to move along with them.

The characteristics defined for this scenario are tackled by auctioning systems. In an auction system, a seller can dynamically sell its services, and buyers compensate them according to how these services fit their requirements. In Section 6.1.3 we study auctioning systems in literature.

6.1.3 Auctioning Systems

Existing solutions in our context are mainly based on two leasing models: fixed and negotiated pricing. In a fixed price system, a seller offers its resources at a specific cost, and the buyers match it. In a negotiated price system, the price of the resource is established by direct competition (auction) between buyers and sellers.

We reviewed existing lease-provisioning systems and identified three types:

- **1 to 1 (Standard lease provisioning):** The conditions of the lease are established beforehand through a Service-Level Agreement (SLA). Examples of this approach can be found in IaaS providers such as Amazon AWS [77] or Google Cloud [121]. This solution is, however, too rigid for dynamic environments. For example, some IaaS Cloud providers charge the allocation of the VM on an hourly basis. That is, users have to pay for slots of 1 hour, even when they might use fractions.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Sellers</td></tr>
<tr><td>1</td><td>N</td></tr>
<tr><td rowspan="2">Buyers</td><td>1</td><td>Standard Lease Provisioning</td><td>Reverse Simple Auction</td></tr>
<tr><td>N</td><td>Direct Simple Auction</td><td>Double Auction</td></tr>
</table>

Table 6.1: Relevant auction systems for leases in distributed clouds

- **1 to N (Simple Auction):** Simple auctions start with a SLA over which the price is negotiated. They are either direct or reverse. In the *direct* approach (one seller, multiple buyers) the SLA is offered beforehand by the seller and the buyer offering the highest price wins the auction. In the *reverse* one (one buyer, multiple sellers) the SLA is offered beforehand by the buyer and the seller requesting the smallest price to fulfill it gains the auction. In a direct 1 to N auction, the seller remains unchanged and values a lease according to the maximum amount payable for it, while in a reverse one an unchanged buyer values it to the minimum pay. A known example of a direct 1 to N approach is Amazon's EC2 Spot Instances [298] and existing works on mobile Clouds such as [299, 300]. This type of lease is less rigid than the standard one. However, these approaches do not encourage competition either between buyers or sellers.
- **N to N (Double Auction):** Double auctions encourage competition both between multiple buyers and sellers. It can be described as a combination of direct and reverse auctions, where the SLA and price are decided according to the market offer [301–303]. Double auctions are the most adaptable solution to the mobile Clouds context, because the conditions are always set dynamically based on the offer and demand.

Table 6.1 summarizes the previous classification of lease-provisioning systems.

All leases, independently from their type, are unfair when one of the buyers possesses more monetary resources than the rest, unbalancing the market. To face this circumstance, several buyers may join efforts to gain an auction against more wealthy opponents, called a group auction. A specific kind of group auctions are group-buying auctions, in which a special price is promised to buyers under the condition of reaching a minimum bid [304]. Bids are either informed (buyers know about the value given to the service by other participants) or blind (no information about other bids are known to buyers). Group auctions are important for applications where multiple buyers collaborate on the same instance. When several users share the same or similar requirements about where to host the VM, they may join efforts through a group auction, in order to host the VM in a node agreed by all.

6.2 Our approach: P2P auction with External Supervision

As described in Chapter 5, the roles of service provider and service manager are unique per microcloud and consume more resources than the rest of nodes. Consequently, they need to be compensated according to their extra work. In order to meet this compensation, the price of a service in an open market should be taken into account. We propose a multi-sided auction system between peered clients (P2P), as we believe that an auction system is a democratic approach to designate where to host computation, fitting all the users involved. Furthermore, it eliminates the problem of free-riding, as the host is being paid.

When the service is launched for the first time the service manager is assigned among the nodes through a 1 to N reverse auction by the base infrastructure. Once assigned, the service manager assigned starts a new auction to select the new service provider. In this new auction, every node blind bids on one or more nodes, as explained in Section 6.2.1. In a blind bid, buyers have no information about other buyers' bids. The use of blind auctioning increases trust from the seller being paid fairly, given that the buyer's perception is not affected by other bids. For example, in a best bid users can negotiate small bids to lower prices. Finally, as the clients move in the network, the service provider may be reallocated when one or more clients decide that, under the current topology, the location of the service provider is not satisfactory enough for them, and request to start a new auction.

6.2.1 Description

In our approach, a common SLA is proposed to the clients, and each node gives certain weight to those characteristics that it considers important, creating a personalized SLA. To ease the auction, sellers expose information about their capacities (information useful for the SLA). These resources include, but are not limited to, CPUs, RAM, Probability of Failure or Energy Efficiency. We define the satisfaction of a node as the difference between what it required and what it gets, similar to other works such as [301,302]. As an example, we show Equation 6.1. Here, the SLA is considered as a linear relation between the trust that the community has of the seller, the fulfillment of the buyer (difference between what it requests and what the seller provides) and the distance between the buyer and seller (delay in ms).

$$
\begin{aligned}
SLA &= Trust * Fulfillment - Distance \\
Fulfillment &= \alpha * (reqPoF - PoF) + \beta * (EE - reqEE) \\
&\quad + \gamma * (RAM - reqRAM) + \delta * (CPUs - reqCPUs) \\
Distance &= \epsilon * \sum (lat_n, i)
\end{aligned}
\tag{6.1}
$$

Trust represents the confidence the system has that the seller is honest about its resources, and is set between 0 and 1 regarding the evaluation other users made of it. *reqPoF, reqEE, reqRAM* and *reqCPUs* represent, respectively, the Probability of Failure - probability of the system failing based on historic, between 0 and 1-, Energy Efficiency - difference between energy consumed while being a client and hosting a service in the historic -, RAM and CPUs requested by the node. *PoF, EE, RAM* and *CPUs* represent the Probability of Failure, Energy Efficiency, RAM and CPUs offered by the node to bid on. α, β, γ, and δ define the relative weight of each variable. lat_n, i represents latency between nodes n and i and ϵ the weight of the distance in the overall SLA. $\alpha, \beta, \gamma, \delta$, and ϵ are configurable by the node, to better fit the users' requirements. This equation can be extended depending on the users' requirements. As stated before, the existence of a trust evaluation is necessary to increase confidence on the seller. If a node does not comply with the accepted SLA (this situation is detected by the rest of the clients), then it will be stripped of the lease without being paid (payment is done after the lease has concluded) and its confidence variable is reduced.

To reduce the possibility of sellers providing fake information about themselves, a recommendation system is implemented, where the history of sellers' performance is evaluated publicly. We extend the idea proposed by Noor et al. [299] through a voting system, where users can evaluate the service received. A node with a poor historical performance receives negative feedback, reducing its possibilities to host new leases. Finally, the existence of a central entity for each microcloud (the service manager) ensures that in the case of an irregularity (such as user's misbehavior or connectivity problems with the node) the lease is canceled and the node hosting the service has the VM taken away and sanctioned, while the auction process is relaunched.

As stated in Chapter 3, a satisfaction threshold needs to be defined as part of the efficiency heuristic. In Chapter 5 we evaluated the use of a fixed threshold for the system. In this chapter we evaluate the use of a dynamic threshold defined by the users. This threshold is based on the users' requirements and measures which SLA (of those provided by the different sellers) satisfies a user. However, given that the auction is a free process where the users pay to receive a service, the result of the auction cannot be neglected. Thus, rather than canceling the result of the auction, the service manager is allowed to participate in the auction to ensure the satisfaction threshold of the clients, using credit provided by the application provider. It can equally align with one or more clients if one of the clients abuses a powerful position over the rest (having significantly more credits than others).

During the auction process, each node bids on one or more sellers. Each user decides which amount to bid to each seller. To do so, users determine a maximum amount to bid on a seller matching all the requirements of their SLA. If a seller does not match all the requirements set by the user, then a proportional amount to the satisfaction of the user (difference between the requirements of the user and what it is offered in the SLA) is placed. Finally, the node with the highest bid obtains the service provider role.

The service manager does not participate during the auction process. After all buyers have made their bids it evaluates the overall satisfaction of the winning result. If it is found that the average satisfaction is negative (that is, if the winning bid results are unsatisfactory for a significant amount of users) then the service manager intervenes. The manager finds the highest bid which provides a positive average satisfaction and adds enough credit to match the former winning bid. This way, the service manager groups with other buyers to obtain a lease which better satisfies the users. While this solution restricts the freedom of the auction process, it ensures that there is no node or a group of buyers which unfairly condition the QoS of the rest of users. The QoS is not only necessary for users, but also for the service owner, which improves the perception users have of its service. This process is described in Algorithm 5.

6.3 Evaluation

The aim of our experiments is to evaluate the performance of different auction mechanisms in a microcloud-based platform operating within a neighborhood, and its effect on a users' incentive scheme. To do so, we focus on users' satisfaction and reward. Experiments have been run using NS3 [63], a packet level network simulator. We used a network of 45 static devices over which a set of mobile devices (between 50 and 100) randomly move, following the same static infrastructure as defined for Chapter 5, in Figure 5.4. On this infrastructure, the same 45-minutes trace of a real shared on-line document as used in Chapter 5 has been reproduced. 15 different auctions occur during the experiments, resulting from the mobile devices' mobility.

On this network we use different bidding strategies for allocating the service provider. For each experiment, all devices in the network (both static and mobile devices) start with

Algorithm 5 Auction's External Supervision.

```
for all nodes do
  Initialize node.bid to 0
for each node in nodes do
  for each neighbor of node do
    {Bid only on those nodes matching its requirements}
    if neighbor matches node requirements then
      node.bid(neighbor, minimumBid
      +(satisfaction)*maxBid/σ)
Initialize currentBid to minimumBid
Initialize currentAvgSatisfaction to −∞
for each node in nodes do
  if node.bid > currentBid then
    if currentAvgSatisfaction >= threshold AND node.AvgSatisfaction <
    threshold then
      currentBid = node.bid − currentBid
    else if currentAvgSatisfaction < threshold AND
    currentAvgSatisfaction < node.AvgSatisfaction
     then
      currentAvgSatisfaction = node.AvgSatisfaction
      chosenNode = node
      currentBid = node.bid
chosenNode.receiveBid
return  chosenNode
```

an initial credit of 10 units. When the LVM is deployed, the Service Manager automatically chooses the node which fits the SLA requirements and requires the least credit. Over time, the mobile devices move along the network, and request a reallocation of the VM which satisfies them better. In our experiments, to ease the comparison of results, we used the SLA described in Equation 6.1, and set the same values for $\alpha = 10, \beta = 10, \gamma = 1, \delta = 1$ and $\epsilon = 1$ on each node.

The bidding formula used to determine the amount of money to bid is described on Equation 6.2.

$$bid = minimumBid + (satisfaction) * maxBid/\sigma \tag{6.2}$$

minimumBid and σ are set by the user. *minimumBid* determines what is the minimum amount to bid and σ the aggressiveness on which the node bids (how fast it increases the amount of money to bid). In our experiments, a *minimumBid* value, in a range between 0 to 10 credits and different for each node, is set for each node. The value $\sigma = 2$ has been used for all clients. To ease comparisons, delay has been considered equal between every connection, so instead of ms, distance has been counted in hops.

The bidding strategies used are:

Standard Double Auction (SDA): Each node bids on its direct neighbors a sum relative to its satisfaction with that node. This strategy rewards low latency over the rest and is the most aggressive, as each node only accepts its maximum satisfaction.

Flexible Double Auction (FDA): It is similar to SDA but besides bidding on its direct neighbors, it also considers distance. To do this, in this auction each node bids on its neighbors. After, it continues bidding in the neighbors' neighbors a smaller percentage of what was bid before. It continues doing this until the distance between the bidding node and the neighbor to be bid on is greater than 10. FDA is less aggressive and less focused on latency than SDA.

Collaborative Double Auction (CDA): As an extension of the FDA, in this strategy every node bids on its direct neighbors and those with a distance of more than 1, but every group - set of clients which share the node to be bid on as a direct neighbor - proposes a common bid. To do so, the node sends a message to all its direct neighbors proposing a new bid. This bid is the result of dividing the amount which the node would bid by itself by the number of direct neighbors. Finally, each nodes bids the minimum between what they were willing to pay and what the neighbors are paying. In this strategy every node uses less credit, while still making a significant contribution. It rewards credit savings for future bids.

Standard Double Auction with External Supervision (SDA-ES): Similar to SDA, but the service manager has the option of matching the biggest bid to provide a better average satisfaction in the system. The interference of the service manager only happens when the average satisfaction is less than the satisfaction threshold. The service provider role is assigned to the node receiving the highest bid with a positive average satisfaction. This strategy ensures that no wealthy buyer controls the outcome of the auction to the detriment of the community (that is, forcing a low satisfaction on the rest).

Random assignment (RND): Assigns the LVM randomly, and it is used as a base line for comparison.

For the experiments, we have set a satisfaction threshold of 5 units, so the average satisfaction of clients is, if possible, bigger than this. This value has been chosen after experimental consideration of different candidates, shown in Figure 6.1. For our experimentation, five different values were considered: $threshold = 0units$, $threshold = 5units$, $threshold = -5units$, $threshold = 10units$ and $threshold = 15units$. Experiments were performed using a variable number of participants. The network topology used is the same as used in Chapter 5. Every experiment represents one hour, during which nodes moved randomly in the network.

Figure 6.1 represents several experiments using the SDA-ES strategy, using different satisfaction thresholds. Each threshold represents an average satisfaction which the service manager considers acceptable. As it is shown in Figure 6.1, the use of a threshold set to 5 units performs well in all the situations. Also, it allows us to see that the addition of external credits in the auction produces, sometimes, feedbacks in the system. For example, for 30 and 40 clients and $threshold = 5units$, the average satisfaction remains well over the threshold. The reason for this feedback to affect positively is that every time an auction is corrected some extra credits are pushed into the system which redistributes the overall wealth in the system.

For the next experiment, shown on Figure 6.2, we evaluated the clients' satisfaction under the different bidding strategies for a variable number of mobile clients. Average, maximum and minimum data are also displayed in Tables 6.2, 6.3 and 6.4.

As depicted, while there is not a great difference between using an SDA or FDA auction, collaborative auctions do not perform well in this scenario - getting, in most cases, a satisfaction close to the one of the random assignment. CDA is less aggressive and more fair between sellers. However, having every user with roughly equal credit eliminates competition. With no competition, there are higher chances of a winning bid which does not satisfy the threshold. Since users are not aggressive in seeking to achieve a satisfactory SLA, the result of the auction is determined by the size of the bidding groups. A large

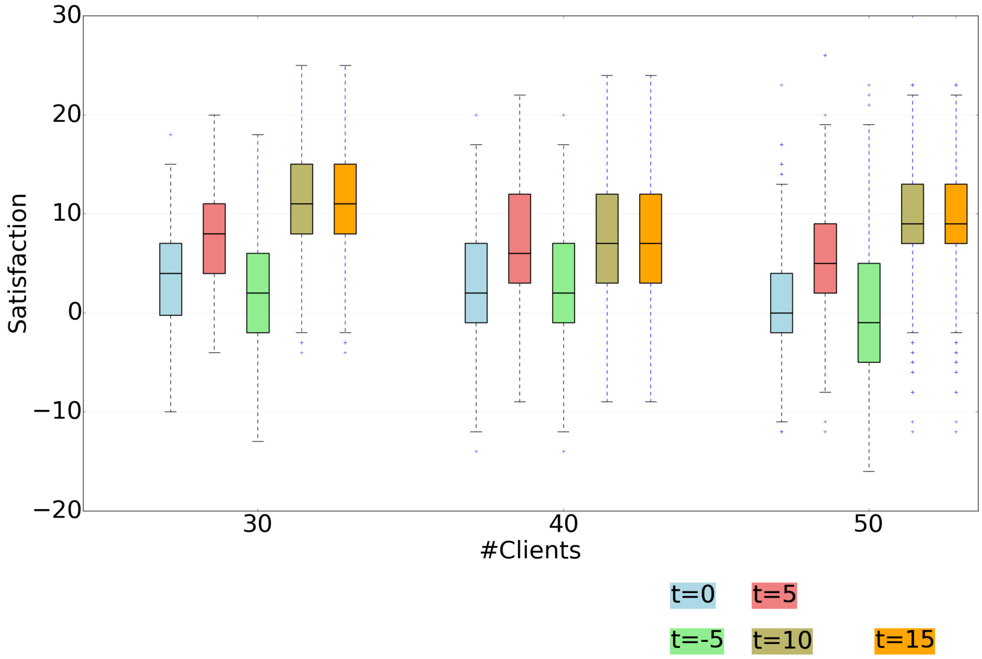


Figure 6.1: Client satisfaction using different threshold values

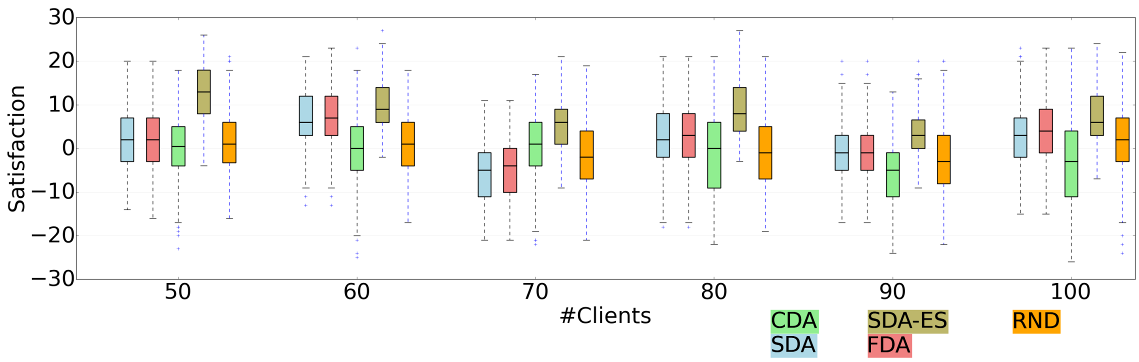


Figure 6.2: Client satisfaction using different strategies

Table 6.2: Average satisfaction per strategy

Strat. \ **#Clients**	**50**	**60**	**70**	**80**	**90**	**100**
SDA	1.62	5.70	-5.39	1.62	-1.31	2.53
FDA	1.62	6.49	-4.37	2.53	-1.31	3.55
CDA	0.15	-0.18	0.60	-0.29	-5.50	-3.46
SDA-ES	12.61	8.76	5.70	7.74	2.76	5.81
RND	0.72	0.49	-2.44	-1.31	-3.4	1.51

group may cause the average satisfaction to be under the threshold. However, under certain circumstances, it may show an equal or better performance than the two others.Finally, SDA-ES performs always better than the rest, due to the external supervision.

Table 6.3: Maximum satisfaction per strategy

Strat. \ #Clients	50	60	70	80	90	100
SDA	19.74	20.65	10.68	20.76	19.63	22.80
FDA	19.74	22.69	10.68	20.76	19.63	22.80
CDA	17.82	22.80	16.69	20.76	12.72	22.58
SDA-ES	25.86	26.88	20.65	26.77	19.63	23.82
RND	20.65	17.71	18.39	20.65	19.63	21.67

Table 6.4: Minimum satisfaction per strategy

Strat. \ #Clients	50	60	70	80	90	100
SDA	-14.40	-13.20	-21.36	-18.30	-17.28	-15.36
FDA	-16.38	-13.20	-21.36	-18.30	-17.28	-15.36
CDA	-23.17	-25.44	-22.26	-22.38	-24.30	-26.34
SDA-ES	-4.37	-2.2	-9.2	-3.40	-9.24	-7.2
RND	-16.42	-17.28	-21.36	-19.31	-22.38	-24.30

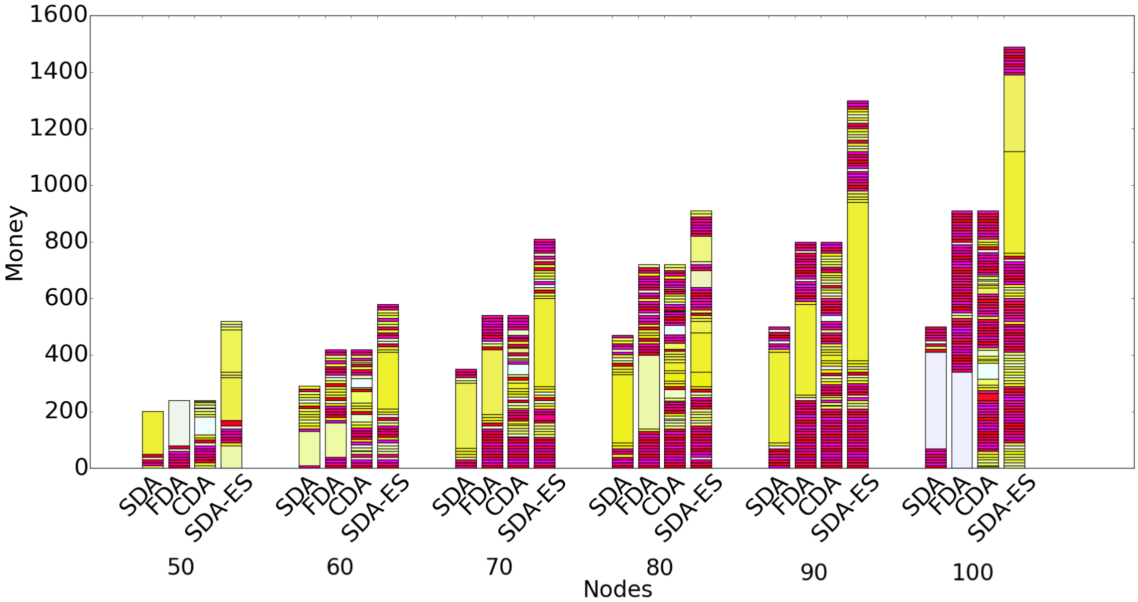
The choice of a threshold of satisfaction in the SDA-ES strategy is of paramount importance. Choosing an unrealistic - too high - satisfaction threshold that no node meets can affect the freedom of credit exchange, as the service manager always decides. On the other hand, a very low threshold is always met and the strategy is the same as SDA.

The distribution of credit among clients participating in the auction at the end of the experiments is shown in Figure 6.3a. In this Figure, each block that compose the bars represents the wealth of a user at the end of the auction. While the distribution of credit among clients is more balanced using a CDA strategy, it also shows that by using a SDA strategy the total amount of credit used in the network is smaller than in any other. This is caused by the nature of the strategy, in which the number of participants in the auction is smaller, as shown in Figure 6.3b. This situation is caused by the unfair distribution of credit, which leads to a small number of clients affecting the result and reducing the number of clients involved in the auction. For the other strategies, we have not observed any difference in the number of clients involved. Last, the SDA-ES strategy shows a better distribution of credit than SDA and FDA but smaller than CDA, making the system more fair.

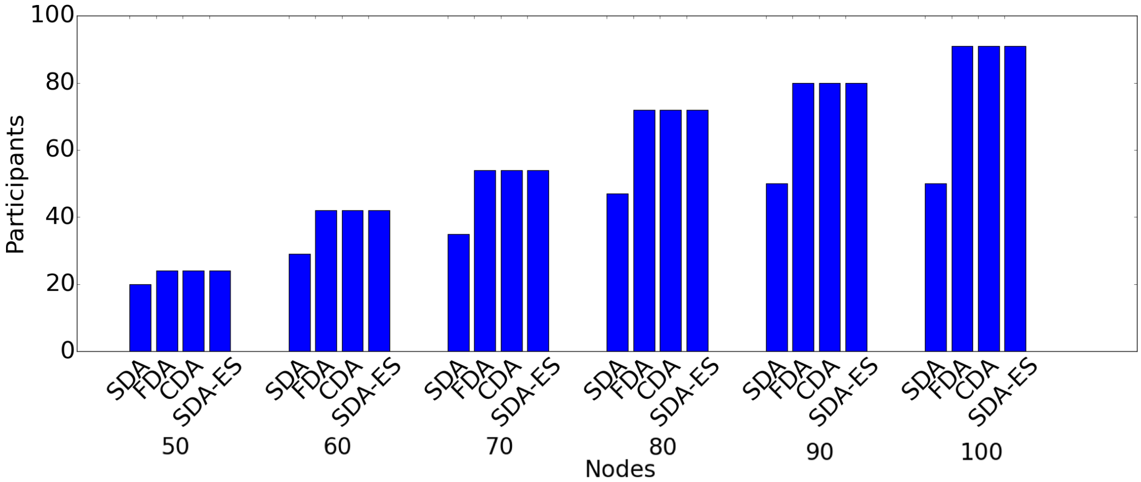
On the one hand, using an SDA-ES strategy also increases the total amount of credit in the system - that is, credit for the users -, through external injections. On the other hand, as stated before, this increase in the credit of the system improves the distribution of credit between users. This situation is shown in Table 6.5, where the highest bid for each strategy is shown. As it can be seen, having more credit in the system leads the buyers to offer higher bids than before.

Table 6.5: Maximum bid per strategy

Strategy \ #Clients	50	60	70	80	90	100
SDA	151	120	230	241	321	341
FDA	160	120	230	260	321	341
CDA	160	120	230	260	321	341
SDA-ES	540	190	300	260	550	350



(a) Credit distribution among clients using different strategies



(b) Participants in the auction using different strategies

Figure 6.3: Credit distribution vs. participants'

6.4 Conclusions

The expansion of geographically located mobile Clouds is a milestone in returning the cloud to users. Furthermore, as mobile technologies advance and become more accessible, it could be the best option to absorb the needed increase in the demand of energy, to reduce unnecessary network traffic and to provide a near to real-time experience. However, to be accepted by users, Mobile Clouds need to be incentivized by the application providers.

Double-auctions are a commonly accepted incentive system in literature. However, double-auction systems do not always provide a fair incentive system in a highly-dynamic and multi-user scenario such as Mobile Clouds. Our approach is able to provide a better solution in these scenarios. We automate a process in which the owner of the service (which, in the end, is the most interested party in the success of the service) has the ability of injecting external credit in the system to avoid abuse of power from wealthy users. We show that, through the use of our system, users receive a service more adapted to their requirements. We also show that this injection of extra credit benefits the competitiveness of the system, as more credit circulating in the system implies a fairer distribution of credit between users.

Chapter 7

Conclusion

This chapter summarizes the contributions of this PhD thesis and presents some future research directions.

7.1 Highlights of the Thesis

The boom in devices connected to the Internet has increased fivefold the amount of traffic in the network in a span of 5 years [11]. The ever-growing base of users means a great increase in demand of energy, which has become a problem of capital importance in today's Internet. Furthermore, nowadays latency experienced by the users exhibits a strong dependence on physical distance to the hosted service. In latency-critical applications distance can be crucial, as it affects the latency.

These issues are, by design, approached by Cloud Computing, due to the ability of virtual machines to be dynamically migrated into heterogeneous infrastructures and different locations. However, centralized Clouds do not follow the design of an elastic and versatile Cloud that is distributed to be available everywhere, like the electricity Grid. In this approach, dominant in today's Internet, the supporting infrastructure on which the platform runs is located in large datacenters. While this approach may offer a better service for applications which need specific and very large computational power, in many others there is no need for the computation to be localized inside a large datacenter.

In order to reduce the Cloud's dependence on such large datacenters, distributed solutions like edge computing are flourishing. In edge computing, virtual machines are hosted in the ISPs infrastructures, which are located near the user. These solutions ensure the ability of the Cloud to keep up with the expected explosion in demand. Moreover, from an energy consumption point of view, datacenters are large constructions which consume large amounts of energy in order to provide a centralized computational power. A reduction in size also reduces the energy they consume. On the other hand, allocating the computation close to the user reduces the physical distance that data has to travel. The smaller the distance traveled by data, the lower the latency perceived by the user.

Finally some other approaches, such as P2P Clouds, contemplate drawing out part of this computation into other nodes involved in the communication process, such as private personal devices. These approaches are able to reduce energy consumption compared to datacenters. Furthermore, they offer a lower latency than other datacenter-centric solutions.

Yet, distributed approaches described in the literature remain inefficient when dealing with latency-critical applications with concurrent users. Most distributed approaches, such as P2P Clouds, rely on having multiple active replicas of the same content. These replicas are computed in multiple nodes which increases the energy consumption, and may cause conflicts between versions (as users work on different versions of the data) and flooding of the network. On the other hand, approaches such as edge computing are still dependent on edge datacenters, and lack the dynamism of elastic approaches such as P2P Clouds.

In this thesis, we show that in these cases where all the clients are localized in a bounded geographical region, the use of a distributed Cloud approach is preferable. In this line, we designed GRaNADA, a software architecture for Cloud computing, which makes use of all available nodes involved in the communication process. In our architecture, communication devices such as ISPs' routers and switches; and domestic routers (also known as "boxes") can be used to host virtual environments. GRaNADA creates an overlay network across these nodes, which we call a microcloud. Every microcloud is responsible for an instance of a service or application. Inside a microcloud, management and hosting of computation are run on different devices, to reduce computational weight on single nodes. However, only one manager and one copy of data are active at the same time.

We show that GRaNADA is more energy-efficient than popular distributed solutions, where data are cached in the ISPs' datacenters. GRaNADA is designed to benefit Internet providers by reducing the traffic along Wide Area Networks (WANs). This way, ISPs can shut down unused resources, which leads towards a reduction of energy consumption and broadband utilization. Furthermore, a reduction in the WAN traffic reduces ISPs' transit costs.

On the other hand, GRaNADA also benefits Cloud providers. Using GRaNADA, they reduce the load in their datacenters and can also shut down unused resources to save energy or use them to accommodate more clients. Finally, by hosting the computation close to the users, latency is reduced, which improves the Quality-of-Service to the user.

While the public network and the datacenter can be used by different users accessing different services, downgrading the configuration of both network and datacenters saves a significant proportion of energy. This downgrade is possible due to the reduction in traffic through specific networks, such as the Wide Area Network. Our experimental results show that through the use of GRaNADA, we save up to 75% of energy consumed by the ISPs, compared with edge computing. Furthermore, GRaNADA is able to use up to 10 times less energy than the worst case scenario of datacenter-centralized approaches. Our experiments were run on a simulation of the French research core network, RENATER. Values are explained by the reduction of the number of hops between users, which decreases the number of network nodes involved in the communication, and the reduction in the involved devices in the datacenter.

Our approach also provides a better experience for the user in terms of latency and adaptability while providing a robust Cloud platform. We evaluated the suitability of GRaNADA in the context of smart cities. We used a scenario which considered a neighborhood-services based platform. Through the definition of this scenario we propose a solution which makes use of our architecture for mobile Cloud computing, where mobile devices offload computation in the Cloud.

Due to their network-based design, microclouds make better use of the network resources than other distributed approaches, reducing the amount of unnecessary data traveling through external networks. We evaluated the RTT, overhead time and robustness of our approach in a platform for neighborhood services. We compared our solution to a mainstream IaaS Cloud provider. We show that in this scenario the use of GRaNADA provides better performance in terms of latency and network awareness than the centralized approach. We also determined the need for an inefficiency heuristic threshold to manage the high dynamism of mobile Clouds. Thanks to this heuristic, the system experiences less migrations, which reduces the use of the computing resources.

Finally, we acknowledge that free riding in a main concern for the users. Indeed, the perception that, in the system, a significant set of users makes use of the services offered by the system without contributing its share may affect the adoption of a distributed technology. We developed a new solution to address this issue. We extended the double auction approach by involving the users as both buyers and sellers of computation. On top of this, we gave the service provider the ability of interfering with the result of the auction, by joining a group of users to ensure fairness in the system. By using a credit-based system, we enhance the perception of the Mobile Cloud as a local infrastructure. On the other hand, it allows users to go beyond the purely digital domain by exchanging this credit by real-life services. It also enhances the adaptability of services to neighborhoods, since larger amounts of credit in a neighborhood increase the number of services and, thus, the credit distribution (for example, a store may accept certain discounts on products as an exchange for credits, which can be returned into the community during the auction process). We show that, compared to other double auction solutions, our system provides a fair distribution of money while still allowing competition between services.

7.2 Future Work

During the work presented in this thesis, some questions arose on which we plan to focus future work. In the first place, we believe that the localized network-aware approach proposed in this thesis can be improved to solve other scenarios than the ones described here. First, it would be interesting to focus on analyzing the suitability of using GRaNADA on different contexts and to address new challenges.

On the one hand, given its geographically localized design, GRaNADA could be instantiated to address the problem of access to the Internet in impoverished and rural areas with limited network infrastructures. More than half of the population still has no connection to the Internet, and only 6% of the population in developing countries has broadband connectivity [305]. Even though the expected number of users connected to the Internet is booming, many rural and impoverished areas lack the infrastructure for their inhabitants to have a useful connection. While some projects aim at improving this infrastructure, such as Internet.org [306], our focus is to provide local services which can be hosted in a light infrastructure, connecting users of the same region. As shown in Chapter 5, GRaNADA can be used to provide neighborhood oriented services to communities with an existing public computing infrastructure. However, in this case we aim at targeting villages with no public infrastructure.

For these regions, services could be created to enhance the interconnection of users, such as localized social network or on-line trading systems. In this case study, users from a impoverished rural area would have access to services connecting them to communities nearby (for example, a shepherd would be able to commerce with neighborhood communities) thanks to a better use to existing resources. While keeping traffic local would reduce the need for expensive communication lines, access from these users to the global Internet would still be restricted to existing infrastructural shortage. Investigating how GRaNADA could be deployed in such case studies and the necessary investment in extra infrastructure would deserve further studies. Also, rural areas have a different density of population than the urban areas considered in Chapter 5. Therefore, it is possible that microclouds cannot be instantaneously formed between users, as there is no connection between them.

In such cases where communication between users is not instantaneous, it is necessary to investigate different ways of sharing data between users, such as delay-tolerant approaches.

On the other hand, GRaNADA can also be implemented to manage emergency situations. In emergency situations, such as earthquakes or wildfires, multiple users need to communicate in real-time to assess damages, retrieve other users, coordinate the response, etcetera. Moreover, in such cases, the communication lines with the external world can be affected. In this line, some solutions have been proposed, such as the use of P2P Cloud networks [307]. However, these distributed solutions suffer from the same downsides which have been shown in this thesis, specially an intensive utilization of the existing broadband. GRaNADA is able to provide services among users of the area, as long as there exists some basic communication infrastructure working among them. Reducing the traffic in the network is important in emergency situations, given that in many of these situations, there is a high chance that the network infrastructure has been affected, reducing the available broadband, and the subsequent chaos is prone to increase the traffic on it. Using a dynamic and distributed architecture which does not impose great stress on the network, such as GRaNADA, would provide a faster and more reliable connection. It would also avoid replication of data, which is prone to cause conflicts of data, and ease the communication between users. In this case study, it would be interesting to investigate how GRaNADA can perform when the local network infrastructure is damaged (thus the resources are shortened) and how fast it can adapt to run on users' devices.

Second, another research direction would consist in using prediction systems to better manage the life-cycle of microclouds. We have shown that the use of thresholds reduces the unnecessary extra computation produced by the changes in the topology caused by the high dynamism of the system. Having predicted information about the expected number of users in an area would reduce unnecessary merge or split of microclouds. For example, determining if a microcloud needs to be split before it becomes too large, based on expected connections.

Finally, the microclouds approach raises security and privacy issues which have not been addressed in this thesis. On the one hand, services running on private devices are prone to attacks from malicious users. Furthermore, users have different levels of expertise in software security. A compromised client machine would be a risk to the microcloud. Some of the multiple security and privacy aspects to consider are man-in-the-middle attacks, compromised hosts, impersonation of roles and pernicious utilization of the light virtual machine from the machine hosting the service provider's role. On the other hand, public infrastructures such as smart cities infrastructures, have implemented lesser physical protection solutions than datacenters [308]. Thus, these resources are easier to be physically accessed from malicious parties. This situation raises a privacy concern, as a device containing private information from the users can be stolen. A thorough study of existing security and privacy breaches in the system and how to adapt them to the system is necessary.

Bibliography

[1] T. Leighton, "Improving performance on the internet," *Communications ACM*, vol. 52, pp. 44–51, Feb. 2009.

[2] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. H. Katz, "Napsac: Design and implementation of a power-proportional web cluster," in *Proceedings of the First ACM SIGCOMM Workshop on Green Networking*, Green Networking '10, (New York, NY, USA), pp. 15–22, ACM, 2010.

[3] B. Aebischer and L. M. Hilty, "The energy demand of ict: A historical perspective and current methodological challenges," in *ICT Innovations for Sustainability* (M. L. Hilty and B. Aebischer, eds.), (Cham), pp. 71–103, Springer International Publishing, 2015.

[4] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *SIGCOMM Computing Communications Review*, vol. 39, pp. 50–55, Dec. 2008.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A view of cloud computing," *Communications ACM*, vol. 53, pp. 50–58, Apr. 2010.

[6] V. Kundra, "State of public sector cloud computing," tech. rep., U.S. Chief Information Officer and the Federal CIO Council, 2012.

[7] A. Aljabre, "Cloud computing for increased business value," *International Journal of Business and Social Science*, vol. 3, no. 1, p. 6, 2012.

[8] RightScale, "2015 state of the cloud report from rightscale." `"https://www.rightscale.com/lp/F2015-state-of-the-cloud-report"`, 2015.

[9] RightScale, "2014 state of the cloud report from rightscale." `"https://www.rightscale.com/lp/F2014-state-of-the-cloud-report"`, 2014.

[10] RightScale, "2016 state of the cloud report from rightscale." `"http://www.rightscale.com/lp/2016-state-of-the-cloud-report?campaign=701700000015euW"`, 2016.

[11] Cisco, "Cisco global cloud index: Forecast and methodology, 20142019." `"http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf"`, 2015.

[12] 451research, "Hosting and cloud study 2015. beyond infrastructure: Cloud 2.0 signifies new opportunities for cloud service providers." `http:`

//download.microsoft.com/download/3/0/B/30B07B25-05CA-46B2-BCF6-1DE845383672/MSFT%20Hosting%20End-User%20Full%20Slide%20Set%203.17.15%20Complete.pdf, 2015.

[13] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, "Trends and challenges in cloud datacenters," *IEEE Cloud Computing*, vol. 1, pp. 10–20, May 2014.

[14] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for internet services," in *31st International Conference on Distributed Computing Systems (ICDCS)*, pp. 131–142, June 2011.

[15] R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, P. Chan, J. Loper, S. Capana, B. Hedman, R. Duff, E. Haines, D. Sass, and A. Fanara, "Report to congress on server and data center energy efficiency: Public law 109-431," tech. rep., Ernest Orlando Lawrence Berkeley National Laboratory, 2008.

[16] Greenpeace, "A guide to building the green internet." "http://www.greenpeace.org/usa/wp-content/uploads/legacy/Global/usa/planet3/PDFs/2015ClickingClean.pdf", 2015.

[17] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos, "Cloud computing: survey on energy efficiency," *ACM Computing Surveys*, vol. Vol. 47, pp. pp. 1–36, January 2015.

[18] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 20152020 white paper," tech. rep., Cisco, 2016. White paper.

[19] A. S. G. Andrae and T. Edler, "On global electricity usage of communication technology: Trends to 2030," *Challenges*, vol. 6, no. 1, p. 117, 2015.

[20] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[21] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, pp. 33–37, Dec. 2007.

[22] E. J. Malecki, *International Handbook of Globalization and World Cities*, ch. 11 Internet networks of world cities: agglomeration and dispersion, pp. 117–125. Edward Elgar Publishing, 2012.

[23] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, and W. Lintner, "United states data center energy usage report," 06/2016 2016.

[24] U. Institute, "2014 data center industry survey." "https://journal.uptimeinstitute.com/2014-data-center-industry-survey/", 2014.

[25] D. Realty, "N. america campos survey results." "http://content.digitalrealty.com/sfc/servlet.shepherd/version/download/06880000000k573", 2013.

[26] M. Bertier, F. Desprez, G. Fedak, A. Lebre, A.-C. Orgerie, J. Pastor, F. Quesnel, J. Rouzaud-Cornabas, and C. Tedeschi, "Beyond the Clouds: How Should Next Generation Utility Computing Infrastructures Be Designed?," in *Cloud Computing* (Z. Mahmood, ed.), Computer Communications and Networks, pp. 325–345, Springer International Publishing, 2014.

[27] T. Kaur and I. Chana, "Energy efficiency techniques in cloud computing: A survey and taxonomy," *ACM Computing Surveys*, vol. 48, pp. 22:1–22:46, Oct. 2015.

[28] H. Alshaer, "An overview of network virtualization and cloud network as a service," *International Journal of Network Management*, vol. 25, no. 1, pp. 1–30, 2015.

[29] M. P. Mills, "The cloud begins with coal: Big data, big networks, big infrastructure, and big power an overview of the electricity used by the global digital ecosystem," tech. rep., Digital Power Group, 2013.

[30] W. V. Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ict electricity consumption from 2007 to 2012," *Computer Communications*, vol. 50, pp. 64 – 76, 2014. Green Networking.

[31] W. Vereecken, W. V. Heddeghem, M. Deruyck, B. Puype, B. Lannoo, W. Joseph, D. Colle, L. Martens, and P. Demeester, "Power consumption in telecommunication networks: overview and reduction strategies," *IEEE Communications Magazine*, vol. 49, pp. 62–69, June 2011.

[32] InternetLiveStats.com, "Internet live stats." `http://www.internetlivestats.com/internet-users/`, 2016.

[33] Cisco, "The zettabyte era: Trends and analysis," tech. rep., Cisco, 2014.

[34] Cisco, "Cisco vni forecast and methodology, 2015-2020," tech. rep., Cisco, 2016. White Paper.

[35] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013.

[36] S. I. Research, "Smb communications pain study: Uncovering the hidden cost of communications barriers and latency." `"https://www.sisinternational.com/smb-communications-pain-study-white-paper-uncovering-the-hidden-cost-of-communications-barriers-and-latency/"`, 2009.

[37] D. Group, "Servers to cloud data centers surpassed a third of overall shipments in first quarter of 2016," tech. rep., DellOro Group, 2016.

[38] Amazon, "Aws global infrastructure." `"http://aws.amazon.com/about-aws/global-infrastructure/"`, 2016.

[39] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review*, vol. 39, no. 1, pp. 68–73, 2008.

[40] D. Hristova, M. Musolesi, and C. Mascolo, "Keep your friends close and your facebook friends closer: A multiplex network approach to the analysis of offline and online social ties," *Computing Research Repository (CoRR)*, vol. abs/1403.8034, 2014.

[41] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *Proceedings of the 18th international conference on World wide web*, 2009.

[42] M. Z. Ahmad and R. Guha, "A tale of nine internet exchange points: Studying path latencies through major regional ixps," in *37th IEEE Conference on Local Computer Networks (LCN)*, pp. 618–625, Oct 2012.

[43] E. Commission, "Seti: Strategic energy technologies information system." `https://setis.ec.europa.eu/set-plan-implementation/technology-roadmaps/european-initiative-smart-cities`, 2016.

[44] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, and M. Mellia, "Energy saving and network performance: A trade-off approach," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, (New York, NY, USA), pp. 41–50, ACM, 2010.

[45] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, (New York, NY, USA), pp. 19–26, ACM, 2003.

[46] S. Puzhavakath Narayanan, Y. S. Nam, A. Sivakumar, B. Chandrasekaran, B. Maggs, and S. Rao, "Reducing latency through page-aware management of web objects by content delivery networks," *SIGMETRICS Performance Evaluation Reviews*, vol. 44, pp. 89–100, June 2016.

[47] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "A hybrid edge-cloud architecture for reducing on-demand gaming latency," *Multimedia Systems*, vol. 20, no. 5, pp. 503–519, 2014.

[48] A. Spurling, "Qos issues for multiplayer gaming," 2005. Case study at Cardiff university 2004.

[49] P. T. Endo, A. V. de Almeida Palhares, N. N. Pereira, G. E. Goncalves, D. Sadok, J. Kelner, B. Melander, and J. E. Mangs, "Resource allocation for distributed cloud: concepts and research challenges," *IEEE Network*, vol. 25, pp. 42–46, July 2011.

[50] I. Narayanan, A. Kansal, A. Sivasubramaniam, B. Urgaonkar, and S. Govindan, "Towards a leaner geo-distributed cloud infrastructure," in *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*, (Philadelphia, PA), USENIX Association, June 2014.

[51] Y. Coady, O. Hohlfeld, J. Kempf, R. McGeer, and S. Schmid, "Distributed cloud computing: Applications, status quo, and challenges," *SIGCOMM Computing Communications Review*, vol. 45, pp. 38–43, Apr. 2015.

[52] W. Lee, T. Kim, S. Kang, and H. Kim, *Revised P2P Data Sharing Scheme over Distributed Cloud Networks*, pp. 165–171. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[53] A. Chandra, J. Weissman, and B. Heintz, "Decentralized edge clouds," *IEEE Internet Computing*, vol. 17, pp. 70–73, Sept 2013.

[54] R. Ranjan and R. Buyya, "Decentralized overlay for federation of enterprise clouds," *Computing Research Repository (CoRR)*, vol. abs/0811.2563, 2008.

[55] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, pp. 37–48, 2009.

[56] S. Islam and J.-C. Grgoire, "Giving users an edge: A flexible cloud model and its application for multimedia," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 823 – 832, 2012.

[57] F. Costa, L. Silva, and M. Dahlin, "Volunteer cloud computing: Mapreduce over the internet," in *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, pp. 1855–1862, May 2011.

[58] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "Seti@home: An experiment in public-resource computing," *Communications ACM*, vol. 45, pp. 56–61, Nov. 2002.

[59] O. Nov, D. Anderson, and O. Arazy, "Volunteer computing: A model of the factors determining contribution to community-based scientific research," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), pp. 741–750, ACM, 2010.

[60] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th International Conference on Supercomputing*, ICS '02, (New York, NY, USA), pp. 84–95, ACM, 2002.

[61] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: Reducing by shutting off cables in bundled links," in *Proceedings of the First ACM SIGCOMM Workshop on Green Networking*, Green Networking '10, (New York, NY, USA), pp. 29–34, ACM, 2010.

[62] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (alr)," *IEEE Transactions on Computers*, vol. 57, pp. 448–461, April 2008.

[63] "Ns3 simulator." `"https://www.nsnam.org/"`, 2015.

[64] I. Cuadrado Cordero, A.-C. Orgerie, and C. A. Morin, "GRaNADA: A Network-Aware and Energy-Efficient PaaS Cloud Architecture," in *IEEE International Conference on Green Computing and Communications (GreenCom)*, (Sydney, Australia), Dec. 2015.

[65] I. Cuadrado Cordero, F. Cuadrado, C. Phillips, A.-C. Orgerie, and C. Morin, "Microcities: a Platform based on Microclouds for Neighborhood Services," in *International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, (Granada, Spain), Dec. 2016.

[66] I. Cuadrado-Cordero, A.-C. Orgerie, and C. Morin, "Incentives for Mobile Cloud Environments through P2P Auctions," in *CloudNet: IEEE International Conference on Cloud Networking*, (Pisa, Italy), Oct. 2016.

[67] P. Mell and T. Grance, "The nist definition of cloud computing recommendations of the national institute of standards and technology," tech. rep., NIST - U.S. National Institute for Standards and Technology, 2011.

[68] J. E. Smith and R. Nair, "The architecture of virtual machines," *Computer*, vol. 38, pp. 32–38, May 2005.

[69] DropboxTM, "Dropbox." `http://www.dropbox.com/`, 2016.

[70] Amazon, "Amazon ec2." `"http://aws.amazon.com/ec2/"`, 2016.

[71] Microsoft, "Microsoft azure." `"https://azure.microsoft.com/en-us/"`.

[72] R. Huang and E. Masanet, "Data center it efficiency measures. the uniform methods project: Methods for determining energy efficiency savings for specific measures." `http://energy.gov/sites/prod/files/2015/01/f19/UMPChapter20-data-center-IT.pdf`, 2015.

[73] J. Barr, "Cloud computing, server utilization and the environment." `"https://aws.amazon.com/blogs/aws/cloud-computing-server-utilization-the-environment/"`, 2015.

[74] E. Star, "Server virtualization." `"https://www.energystar.gov/products/low_carbon_it_campaign/12_ways_save_energy_data_center/server_virtualization"`, 2013.

[75] T. S. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 27–33, IEEE Computer Society, 2010.

[76] A. Ghobadi, R. Karimi, F. Heidari, and M. Samadi, "Cloud computing, reliability and security issue," in *16th International Conference on Advanced Communication Technology*, pp. 504–511, IEEE, 2014.

[77] Amazon, "Amazon cloud infrastructure." `http://aws.amazon.com/about-aws/global-infrastructure/`, 2016.

[78] Google, "Google app engine." `"https://cloud.google.com/appengine/docs"`, 2016.

[79] Google, "Google drive." `"https://www.google.com/drive/"`, 2016.

[80] Google, "Google drive." `"mail.google.com"`, 2016.

[81] Microsoft, "Microsoft one drive." `"https://onedrive.live.com/"`, 2016.

[82] IBM, "Softlayer." œ`http://www.softlayer.com/info/privacy`, 2016.

[83] Avere, "Google cloud - avere hybrid cloud nas." `"https://cloud.google.com/partners/pages/avere-systems/"`, 2016.

[84] Microsoft, "Microsoft hybrid cloud platform." `"https://www.microsoft.com/en-us/cloud-platform/hybrid-cloud"`, 2016.

[85] Amazon, "Amazon virtual private cloud (vpc)." œ`https://aws.amazon.com/vpc/`œ.

[86] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "Interconnecting federated clouds by using publish-subscribe service," *Cluster Computing*, vol. 16, no. 4, pp. 887–903, 2013.

[87] P. Harsh, J. Yvon, R. G. Cascella, and C. Morin, "Contrail Virtual Execution Platform - Challenges in Being Part of a Cloud Federation," in *4th European Conference Servicewave 2011*, (Poznan, Poland), Oct. 2011.

[88] S. Zhang, S. Zhang, X. Chen, and X. Huo, "Cloud computing research and development trend," in *Second International Conference on Future Networks, 2010. ICFN '10*, pp. 93–97, Jan 2010.

[89] D. Merkel, "Docker: Lightweight linux containers for consistent development and deployment," *Linux J.*, vol. 2014, Mar. 2014.

[90] Adobe, "Adobe premiere." `http://www.adobe.com/products/premiere-elements.html`, 2016.

[91] Google, "Picasa." `picasaweb.google.com`, 2016.

[92] Google, "Google docs." `docs.google.com`, 2016.

[93] TwistedWave, "twistedwave.com." `twistedwave.com`, 2016.

[94] D. Hatley and I. Pirbhai, *Strategies for real-time system specification.* Addison-Wesley, 2013.

[95] Google, "Google cloud cdn." `"https://cloud.google.com/cdn/docs/overview"`, 2016.

[96] Amazon, "Amazon cloudfront content delivery network (cdn)." `"https://aws.amazon.com/cloudfront/"`, 2016.

[97] B. Inc., "Bittorrent." `"http://www.bittorrent.com/"`, 2016.

[98] DropboxTM, "Dropbox plans." `https://www.dropbox.com/business/plans-comparison`, 2016.

[99] T. Publishing, "Record breaking torrents." `http://https://torrentfreak.com/game-thrones-makes-history-200000-strong-torrent-swarm-140506/`, 2014.

[100] T. Publishing, "Distribution of users in popular torrents." `https://torrentfreak.com/game-of-thrones-premiere-triggers-piracy-craze-140407/`, 2014.

[101] i. DevHD, "Feedly." `feedly.com/`, 2016.

[102] Facebook, "Facebook." `www.facebook.com`, 2016.

[103] RiotGames, "League of legends." `http://na.leagueoflegends.com/`, 2016.

[104] Google, "Youtube." `www.youtube.com`, 2016.

[105] Netfix, "Netflix." `www.netflix.com`, 2016.

[106] Google, "Youtube official blog." `https://youtube.googleblog.com/2016/09/with-nearly-2-million-concurrent.html`, 2016.

[107] R. Prodan, S. Ostermann, and M. Sperk, "Evaluating high-performance computing on google app engine," *IEEE Software*, vol. 29, no. undefined, pp. 52–58, 2011.

[108] R. N. Calheiros, R. Ranjan, and R. Buyya, "Virtual machine provisioning based on analytical performance and qos in cloud computing environments," in *2011 International Conference on Parallel Processing*, pp. 295–304, IEEE, 2011.

[109] E. F. Coutinho, F. R. de Carvalho Sousa, P. A. L. Rego, D. G. Gomes, and J. N. de Souza, "Elasticity in cloud computing: a survey," *annals of telecommunications - annales des télécommunications*, vol. 70, no. 7, pp. 289–309, 2015.

[110] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On technical security issues in cloud computing," in *2009 IEEE International Conference on Cloud Computing*, pp. 109–116, Sept 2009.

[111] I. Hussain, S. Djahel, Z. Zhang, and F. Naït-Abdesselam, "A comprehensive study of flooding attack consequences and countermeasures in session initiation protocol (sip)," *Security and Communication Networks*, vol. 8, no. 18, pp. 4436–4451, 2015.

[112] S. Buchegger, "Delay-tolerant social networking," in *1st extreme workshop in communications*, 2009.

[113] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '09, (New York, NY, USA), pp. 299–308, ACM, 2009.

[114] M. Bolhuis, "A comparison between centralized and distributed cloud storage datacenter topologies," in *19th Twente Student Conference on IT*, June 2013.

[115] M. Alicherry and T. V. Lakshman, "Network aware resource allocation in distributed clouds," in *Proceedings of IEEE INFOCOM 12*, pp. 963–971, March 2012.

[116] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside dropbox: Understanding personal cloud storage services," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, (New York, NY, USA), pp. 481–494, ACM, 2012.

[117] H. Slatman, "Opening up the sky: a comparison of performance-enhancing features in skydrive and dropbox," in *Proceedings of the 18th Twente Student Conference on IT*, 2013.

[118] O. Babaoglu, M. Marzolla, and M. Tamburini, "Design and implementation of a p2p cloud system," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, (New York, NY, USA), pp. 412–417, ACM, 2012.

[119] Z. Yang, B. Y. Zhao, Y. Xing, S. Ding, F. Xiao, and Y. Dai, "Amazingstore: available, low-cost online storage service using cloudlets.," in *International Workshop in Peer to Peer Systems (IPTPS)*, vol. 10, pp. 2–2, 2010.

[120] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys*, vol. 46, pp. 47:1–47:31, Mar. 2014.

[121] GoogleTM, "Google cloud." `https://cloud.google.com/`, 2016.

[122] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 732 – 749, 2011.

[123] O. Babaoglu and M. Marzolla, "The people's cloud," *IEEE Spectrum*, vol. 51, pp. 50–55, October 2014.

[124] K. Chard, K. Bubendorfer, S. Caton, and O. Rana, "Social Cloud Computing: A Vision for Socially Motivated Resource Sharing," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 551–563, 2012.

[125] M. Steiner, B. G. Gaglianello, V. Gurbani, V. Hilt, W. Roome, M. Scharf, and T. Voith, "Network-aware service placement in a distributed cloud environment," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, (New York, NY, USA), pp. 73–74, ACM, 2012.

[126] B. Solomon, D. Ionescu, C. Gadea, S. Veres, M. Litoiu, and J. Ng, "Distributed clouds for collaborative applications," in *International Conference on Collaboration Technologies and Systems (CTS)*, pp. 218–225, May 2012.

[127] B. Mejias and P. Roy, "From mini-clouds to cloud computing," in *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop (SASOW)*, pp. 234–238, Sept 2010.

[128] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of internet content delivery systems," *OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation*, vol. 36, pp. 315–327, Dec. 2002.

[129] S. Bakiras and T. Loukopoulos, "Combining replica placement and caching techniques in content distribution networks," *Computer Communications*, vol. 28, no. 9, pp. 1062 – 1073, 2005.

[130] A. Passarella, "A survey on content-centric technologies for the current internet: {CDN} and {P2P} solutions," *Computer Communications*, vol. 35, no. 1, pp. 1 – 32, 2012.

[131] A. Technologies, "Akamai technologies." `https://www.akamai.com`, 2016.

[132] H. Hu, Y. Wen, T. S. Chua, Z. Wang, J. Huang, W. Zhu, and D. Wu, "Community based effective social video contents placement in cloud centric cdn network," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, July 2014.

[133] G. Pierre, M. van Steen, M. Szymaniak, and S. Sivasubramanian, *Global Data Management*, ch. From Web Servers to Ubiquitous Content Delivery, pp. 324–341. No. ISBN 1-58603-629-7, IOS Press, july 2006. `http://www.iospress.nl/loadtop/load.php?isbn=1586036297`.

[134] F. Marozzo, D. Talia, and P. Trunfio, "P2p-mapreduce: Parallel data processing in dynamic cloud environments," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1382 – 1402, 2012. JCSS Special Issue: Cloud Computing 2011.

[135] K. Xu, M. Song, X. Zhang, and J. Song, "A cloud computing platform based on p2p," in *IEEE International Symposium on IT in Medicine Education. ITIME '09*, vol. 1, pp. 427–432, Aug 2009.

[136] Z. Peng, H. Ting-lei, L. Cai-xia, and X. Wang, "Research of p2p architecture based on cloud computing," in *2010 International Conference on Intelligent Computing and Integrated Systems (ICISS)*, pp. 652–655, Oct 2010.

[137] X. Jin and Y. K. Kwok, "Cloud assisted p2p media streaming for bandwidth constrained mobile subscribers," in *2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 800–805, Dec 2010.

[138] Z. Li, T. Zhang, Y. Huang, Z.-L. Zhang, and Y. Dai, "Maximizing the bandwidth multiplier effect for hybrid cloud-p2p content distribution," in *Proceedings of the 2012 IEEE 20th International Workshop on Quality of Service*, IWQoS '12, (Piscataway, NJ, USA), pp. 20:1–20:9, IEEE Press, 2012.

[139] D. Hughes, G. Coulson, and J. Walkerdine, "Free riding on gnutella revisited: The bell tolls?," *IEEE Distributed Systems Online*, vol. 6, pp. 1–, June 2005.

[140] P. Rodriguez, S.-M. Tan, and C. Gkantsidis, "On the feasibility of commercial, legal p2p content distribution," *SIGCOMM Computing Communications Review*, vol. 36, pp. 75–78, Jan. 2006.

[141] A. Chandra and J. B. Weissman, "Nebulas: Using distributed voluntary resources to build clouds.," in *HotCloud*, 2009.

[142] M. Ryden, K. Oh, A. Chandra, and J. Weissman, "Nebula: Distributed edge cloud for data intensive computing," in *2014 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 57–66, March 2014.

[143] G. Huerta-Canepa and D. Lee, "A Virtual Cloud Computing Provider for Mobile Devices," in *ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, pp. 6:1–6:5, ACM, 2010.

[144] M. Satyanarayanan, "Mobile Computing: The Next Decade," in *ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, pp. 5:1–5:6, ACM, 2010.

[145] E. Marinelli, "Hyrax: Cloud Computing on Mobile Devices using MapReduce," Master's thesis, Carnegie Mellon University, 2009.

[146] O. Amft and P. Lukowicz, "From Backpacks to Smartphones: Past, Present, and Future of Wearable Computers," *IEEE Pervasive Computing*, vol. 8, pp. 8–13, July 2009.

[147] X. Luo, "From Augmented Reality to Augmented Computing: A Look at Cloud-Mobile Convergence," in *IEEE International Symposium on Ubiquitous Virtual Reality (ISURV)*, pp. 29–32, July 2009.

[148] S. Islam and J.-C. Grgoire, "Network edge intelligence for the emerging next-generation internet," *Future Internet*, vol. 2, no. 4, p. 603, 2010.

[149] A. Lebre, J. Pastor, and . The DISCOVERY Consortium, "The DISCOVERY Initiative - Overcoming Major Limitations of Traditional Server-Centric Clouds by Operating Massively Distributed IaaS Facilities," Research Report RR-8779, Inria, Sept. 2015.

[150] F. Bonomi, "Connected vehicles, the internet of things, and fog computing," in *The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA*, pp. 13–15, 2011.

[151] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[152] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *SIGCOMM Computing Communications Review*, vol. 44, pp. 27–32, Oct. 2014.

[153] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *CoRR*, vol. abs/1602.00484, 2016.

[154] F. Jalali, R. Ayre, A. Vishwanath, K. Hinton, T. Alpcan, and R. S. Tucker, "Energy consumption of content distribution from nano data centers versus centralized data centers," *SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 49–54, 2014.

[155] R. Krishnan, M. D. Smith, Z. Tang, and R. Telang, "The impact of free-riding on peer-to-peer networks," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp. 10 pp.–, Jan 2004.

[156] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "You better be honest: Discouraging free-riding and false-reporting in mobile crowdsourcing," in *2014 IEEE Global Communications Conference*, pp. 4971–4976, Dec 2014.

[157] K. Ranganathan and I. Foster, "Identifying dynamic replication strategies for a high-performance data grid," *Grid ComputingGRID 2001*, pp. 75–86, 2001.

[158] A. Fedoruk and R. Deters, "Improving fault-tolerance by replicating agents," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pp. 737–744, ACM, 2002.

[159] D. J. Abadi, "Consistency tradeoffs in modern distributed database system design," *Computer-IEEE Computer Magazine*, vol. 45, no. 2, p. 37, 2012.

[160] D. Li, A. Desai, Z. Yang, K. Mueller, S. Morris, and D. Stavisky, "Multicast cloud with integrated multicast and unicast content distribution routing," in *Web Content Caching and Distribution: Proceedings of the 8th International Workshop* (F. Douglis and B. D. Davison, eds.), (Dordrecht), pp. 109–118, Springer Netherlands, 2004.

[161] F. Martignon, S. Paris, I. Filippini, L. Chen, and A. Capone, "Efficient and truthful bandwidth allocation in wireless mesh community networks," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 161–174, Feb 2015.

[162] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 346–351, April 2014.

[163] "O3b networks." `"http://www.o3bnetworks.com/wp-content/uploads/2015/02/white-paper_latency-matters.pdf"`, 2015.

[164] D. Delaney, T. Ward, and S. McLoone, "On consistency and network latency in distributed interactive applications: A survey–part i," *Presence: Teleoper. Virtual Environ.*, vol. 15, pp. 218–234, Apr. 2006.

[165] J. Brutlag, "Speed matters for google web search," *Google. June*, 2009.

[166] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft, "Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades," in *International Conference on World Wide Web (WWW)*, pp. 457–466, 2011.

[167] A. Sanike, A. Subramanyam, S. S. S. Reddy, and G. RaghuRam, "Load balancing technique to handle the congestion in the communication networks," in *2015 Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG)*, pp. 289–293, Dec 2015.

[168] T. D. BUREAU, "Ict quality of service regulation: Practices and proposals," tech. rep., INTERNATIONAL TELECOMMUNICATION UNION, 2006.

[169] W. B. Norton, "Internet service providers and peering," in *Proceedings of NANOG*, vol. 19, pp. 1–17, 2001.

[170] J. Miettinen, V. Pierne, T. Myren, J. Hopia, A. Todosijevic, V. Ajanovski, R. Yoshinov, C. Friacas, V. Gazivoda, M. Cabak, *et al.*, "Deliverable d3. 2 annual report on campus best practice," tech. rep., GEANT, 2016.

[171] I. T. Union, "Domestic enforcement of telecommunication laws, rules, and regulations by national telecommunications regulatory authorities." `www.itu.int/dms_pub/itu-d/opb/stg/D-STG-SG01.18.1-2010-MSW-E.docx`, 2013.

[172] A. Bandyopadhyay, S. Vuppala, and P. Choudhury, "A simulation analysis of flooding attack in manet using ns-3," in *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, pp. 1–5, Feb 2011.

[173] Y. Sun, D. Liu, Y. Yao, and C. Phillips, "Adaptive packet aggregation scheduling scheme for crmanet," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pp. 1019–1023, Sept 2014.

[174] M. Zheleva, P. Schmitt, M. Vigil, and E. Belding, "Internet bandwidth upgrade: implications on performance and usage in rural zambia," *Information Technologies & International Development*, vol. 11, no. 2, pp. pp–1, 2015.

[175] N. W. Group, "BGP Version 4." `https://tools.ietf.org/html/rfc4271`, 2016.

[176] Cisco, "OSPF Version 2." `http://www.cisco.com/c/en/us/tech/ip/open-shortest-path-first-ospf/index.html`, 2016.

[177] J. Vallet and O. Brun, "Online ospf weights optimization in ip networks," *Computer Networks*, vol. 60, pp. 1–12, 2014.

[178] Greenpeace, "How dirty is your data? a look at the energy choices that power cloud computing." `http://www.greenpeace.org/international/Global/international/publications/climate/2011/Cool%20IT/dirty-data-report-greenpeace.pdf`, 2011.

[179] E. Gelenbe and R. Lent, "Trade-offs between energy and quality of service," in *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pp. 1–5, Oct 2012.

[180] A.-C. Orgerie and L. Lefevre, "Energy-efficient framework for networks of large-scale distributed systems," *2014 IEEE International Symposium on Parallel and Distributed Processing with Applications*, vol. 0, pp. 250–255, 2011.

[181] A. B. B. A. Group, "Powering the cloud." `"http://www04.abb.com/global/seitp/seitp202.nsf/0/7d3fc81ec63e658dc12578940027daea//$file/ABB_DataCenter_infographic_v0.9.pdf"`, 2012.

[182] N. R. D. C. (NRDC), "Data center efficiency assessment." `https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IP.pdf`, 2014.

[183] Accenture, "Cloud computing and sustainability: The environmental benefits of moving to the cloud." `œhttp://gesi.org/files/Reports/AssessmentMethodologyCasteStudy_CloudComputingSustainability-Nov2010.pdf`, 2010.

[184] J. Buysse, K. Georgakilas, A. Tzanakaki, M. D. Leenheer, B. Dhoedt, C. Develder, and P. Demeester, "Calculating the minimum bounds of energy consumption for cloud networks," in *Proceedings of 20th International Conference on Computer Communications and Networks, ICCCN 2011, Hawaii* (H. Wang, J. Li, G. N. Rouskas, and X. Zhou, eds.), pp. 1–7, IEEE, 2011.

[185] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755 – 768, 2012.

[186] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, pp. 149–167, Jan 2011.

[187] T. Heath, B. Diniz, E. V. Carrera, W. Meira, and R. Bianchini, "Energy conservation in heterogeneous server clusters," *Energy conservation in heterogeneous server clusters. In PPoPP*, vol. 5, pp. 186–195, 2005.

[188] H. Sun, P. Stolf, J. Pierson, and G. D. Costa, "Energy-efficient and thermal-aware resource management for heterogeneous datacenters," *CoRR*, vol. abs/1410.3104, 2014.

[189] C. Mobius, W. Dargie, and A. Schill, "Power consumption estimation models for processors, virtual machines, and servers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 1600–1614, June 2014.

[190] A.-C. Orgerie, L. Lefevre, I. Guerin-Lassous, and D. M. L. Pacheco, "ECOFEN: An End-to-end Energy Cost mOdel and Simulator For Evaluating Power Consumption

in Large-scale Networks," in *Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WOWMOM '11, (Washington, DC, USA), pp. 1–6, IEEE Computer Society, 2011.

[191] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, "Energy consumption of telecommunication networks and related improvement options," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, pp. 285–295, March 2011.

[192] K. Gomez, D. Boru, R. Riggio, T. Rasheed, D. Miorandi, and F. Granelli, "Measurement-based modelling of power consumption at wireless access network gateways," *Computer Networks*, vol. 56, no. 10, pp. 2506 – 2521, 2012.

[193] A. Gazo Cervero, M. Chincoli, L. Dittmann, A. Fischer, A. E. Garcia, J. Galn-Jimnez, L. Lefevre, H. de Meer, T. Monteil, P. Monti, A.-C. Orgerie, L.-F. Pau, C. Phillips, S. Ricciardi, R. Sharrock, P. Stolf, T. Trinh, and L. Valcarenghi, *Large-scale Distributed Systems and Energy Efficiency: A Holistic View*, ch. 3. Green Wired Networks, pp. 263–286. Wiley Series on Parallel and Distributed Computing, Wiley, 2000.

[194] Apple, "Mac pro: Power consumption and thermal output (btu/h) information." `https://support.apple.com/en-us/HT201796`, 2016.

[195] HP, "Hp probook 6570b notebook pc - specifications." `http://h20564.www2.hp.com/hpsc/doc/public/display?docId=emr_na-c03370081`, 2016.

[196] E. N. Power, "Energy logic: Reducing data center energy consumption by creating savings that cascade across systems." `"http://viewer.media.bitpipe.com/1246912269_236/1264116198_198/Liebert_EnergyLogic.pdf"`, 2008.

[197] S. C. Bhattacharyya and G. R. Timilsina, "Energy demand models for policy formulation: a comparative study of energy demand models," *World Bank Policy Research Working Paper Series, Vol*, 2009.

[198] A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas, "Demystifying energy consumption in grids and clouds," in *Green Computing Conference*, pp. 335–342, IEEE, 2010.

[199] G. Meyer, "Cold aisle/hot aisle containment system for computer servers in a data center," Jan. 12 2012. US Patent App. 13/135,452.

[200] V. A. John Riemann, Kevin Brown, "Hot air vs. cold air containment for datacenters," tech. rep., Data Center Science Center, 2010.

[201] G. Shen, Y. Lui, and S. K. Bose, "Follow the sun, follow the wind lightpath virtual topology reconfiguration in ip over wdm network," *Journal of Lightwave Technology*, vol. 32, pp. 2094–2105, June 2014.

[202] W. V. Heddeghem, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester, "Distributed computing for carbon footprint reduction by exploiting low-footprint energy availability," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 405 – 414, 2012.

[203] Google, "Haminas google datacenters." `"http://www.google.com/about/datacenters/inside/locations/hamina/"`, 2016.

[204] E. Star, "Energy savings from energy star-qualified servers." "http://www.energystar.gov/ia/products/downloads/ES_server_case_study.pdf", 2010.

[205] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor, "Storage modeling for power estimation," in *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, p. 3, ACM, 2009.

[206] G. Rizzelli, A. Morea, M. Tornatore, and O. Rival, "Energy efficient traffic-aware design of onoff multi-layer translucent optical networks," *Computer Networks*, vol. 56, no. 10, pp. 2443 – 2455, 2012. Green communication networks.

[207] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *SIGARCH Comput. Archit. News*, vol. 35, pp. 13–23, June 2007.

[208] G. Chen, R. Shetty, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and M. Wolczko, "Tuning garbage collection for reducing memory system energy in an embedded java environment," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 1, no. 1, pp. 27–55, 2002.

[209] A. Heydari, S. H. Park, and A. M. Michael, "Cooling servers in a data center using fans external to servers," June 2 2011. US Patent App. 12/628,143.

[210] L. Lefèvre and A.-C. Orgerie, "Designing and evaluating an energy efficient cloud," *The Journal of Supercomputing*, vol. 51, no. 3, pp. 352–373, 2010.

[211] J. A. Gallego Arrubla, Y. M. Ko, R. J. Polansky, E. Pérez, L. Ntaimo, and N. Gautam, "Integrating virtualization, speed scaling, and powering on/off servers in data centers for energy efficiency," *IIE Transactions*, vol. 45, no. 10, pp. 1114–1136, 2013.

[212] A. Gandhi, M. Harchol-Balter, and M. A. Kozuch, "Are sleep states effective in data centers?," in *Green Computing Conference (IGCC), 2012 International*, pp. 1–10, June 2012.

[213] V. Villebonnet, G. Da Costa, L. Lefèvre, J.-M. Pierson, and P. Stolf, ""Big, Medium, Little": Reaching Energy Proportionality with Heterogeneous Computing Scheduler," *Parallel Processing Letters*, vol. 25, no. 03, p. 1541006, 2015.

[214] A. Vahdat, A. Lebeck, and C. S. Ellis, "Every joule is precious: The case for revisiting operating system design for energy efficiency," in *Proceedings of the 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System*, EW 9, (New York, NY, USA), pp. 31–36, ACM, 2000.

[215] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob, S. Yada, S. Marella, P. Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T. Lund-Larsen, S. Steibl, S. Borkar, V. De, R. V. D. Wijngaart, and T. Mattson, "A 48-core ia-32 message-passing processor with dvfs in 45nm cmos," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 108–109, Feb 2010.

[216] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *SIGARCH Computing Architectures News*, vol. 35, pp. 13–23, June 2007.

[217] G. Eads, J. Colmenares, S. Hofmeyr, S. Bird, D. Bartolini, D. Chou, B. Glutzman, K. Asanovi, and J. D. Kubiatowicz, “Building an adaptive operating system for predictability and efficiency,” Tech. Rep. UCB/EECS-2014-137, EECS Department, University of California, Berkeley, Jul 2014.

[218] E. Pinheiro and R. Bianchini, “Energy conservation techniques for disk array-based servers,” in *ACM International Conference on Supercomputing 25th Anniversary Volume*, pp. 369–379, ACM, 2014.

[219] J. Torres, D. Carrera, K. Hogan, R. Gavaldà, V. Beltran, and N. Poggi, “Reducing wasted resources to help achieve green data centers,” in *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, pp. 1–8, IEEE, 2008.

[220] C. H. Hsu, S. C. Chen, C. C. Lee, H. Y. Chang, K. C. Lai, K. C. Li, and C. Rong, “Energy-aware task consolidation technique for cloud computing,” in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, pp. 115–121, Nov 2011.

[221] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, “Optimality analysis of energy-performance trade-off for server farm management,” *Performance Evaluation*, vol. 67, no. 11, pp. 1155 – 1171, 2010.

[222] M. B. Srivastava, A. P. Chandrakasan, and R. W. Brodersen, “Predictive system shutdown and other architectural techniques for energy efficient programmable computation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 4, pp. 42–55, March 1996.

[223] A. Muhammad, P. Monti, I. Cerutti, L. Wosinska, P. Castoldi, and A. Tzanakaki, “Energy-efficient wdm network planning with dedicated protection resources in sleep mode,” in *GLOBECOM*, pp. 1–5, IEEE, 2010.

[224] D. Meisner, B. T. Gold, and T. F. Wenisch, “Powernap: Eliminating server idle power,” *SIGARCH Computing Architectures News*, vol. 37, pp. 205–216, Mar. 2009.

[225] Cisco, “Multi-topology routing.” `http://www.cisco.com/c/en/us/td/docs/ios/12_2sr/12_2srb/feature/guide/srmtrdoc.html`, 2006.

[226] C. Gunaratne, K. Christensen, and B. Nordman, “Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed,” *International Journal in Network Management*, vol. 15, pp. 297–310, Sept. 2005.

[227] P. Reviriego, J.-A. Hernadez, D. Larrabeiti, and J. A. Maestro, “Burst transmission for energy-efficient ethernet,” *IEEE Internet Computing*, vol. 14, pp. 50–57, July 2010.

[228] B. Eckermann, “Low-power network standby in the home and office,” tech. rep., Digital Networking, 2011.

[229] R. Bolla, R. Bruschi, A. Cianfrani, and M. Listanti, “Enabling backbone networks to sleep,” *IEEE Network*, vol. 25, pp. 26–31, March 2011.

[230] IEEE, “Ieee 802.3az task force group.” `"http://www.ieee802.org/3/az/public/index.html"`, 2010.

[231] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, "Ieee 802.3az: the road to energy efficient ethernet," *IEEE Communications Magazine*, vol. 48, pp. 50–56, November 2010.

[232] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware networks: Reducing power consumption by switching off network elements," in *FEDERICA-Phosphorus tutorial and workshop (TNC2008)*, Citeseer, 2008.

[233] H. Mellah, "Review of facts, data and proposals for a greener internet," in *In Proceedings of Broadnets09*, 2009.

[234] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[235] W. Guo and T. OFarrell, "Reducing energy consumption of wireless communications," in *UK Research Council Conference on Digital Engagement 2011*, Citeseer, 2011.

[236] A. Gladisch, C. Lange, and R. Leppla, "Power efficiency of optical versus electronic access networks," in *34th European Conference on Optical Communication*, pp. 1–1, Sept 2008.

[237] G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, "Optical switching: switch fabrics, techniques, and architectures," *Journal of Lightwave Technology*, vol. 21, pp. 384–405, Feb 2003.

[238] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, (Berkeley, CA, USA), pp. 323–336, USENIX Association, 2008.

[239] S. Raman, K. Veezhinathan, B. Venkat, and G. Raina, "Using timers to switch-off tcam banks in routers," in *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, Dec 2013.

[240] V. Soteriou and L.-S. Peh, "Dynamic power management for power optimization of interconnection networks using on/off links," in *Proceedings of the 11th Symposium on High Performance Interconnects, 2003.*, pp. 15–20, Aug 2003.

[241] L. Chiaraviglio, A. Cianfrani, A. Coiro, M. Listanti, and M. Polverini, "Green routing/switching and transport," *Green Communications: Principles, Concepts and Practice*, pp. 257–276, 2015.

[242] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *2009 IEEE International Conference on Communications*, pp. 1–6, June 2009.

[243] B. G. Assefa and O. Ozkasap, "State-of-the-art energy efficiency approaches in software defined networking," *ICN 2015*, p. 268, 2015.

[244] X. Xiao, A. Hannan, B. Bailey, and L. M. Ni, "Traffic engineering with mpls in the internet," *IEEE Network*, vol. 14, pp. 28–33, Mar 2000.

[245] N. Vasić and D. Kostić, "Energy-aware traffic engineering," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, (New York, NY, USA), pp. 169–178, ACM, 2010.

[246] E. Amaldi, A. Capone, and L. Gianoli, "Energy-aware {IP} traffic engineering with shortest path routing," *Computer Networks*, vol. 57, no. 6, pp. 1503 – 1517, 2013.

[247] M. A. Adnan, R. Sugihara, and R. K. Gupta, "Energy efficient geographical load balancing via dynamic deferral of workload," in *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, CLOUD '12, (Washington, DC, USA), pp. 188–195, IEEE Computer Society, 2012.

[248] M. Rifai, D. L. Pacheco, and G. Urvoy-Keller, "Towards enabling green routing services in real networks," in *2014 IEEE Online Conference on Green Communications (OnlineGreenComm)*, pp. 1–7, Nov 2014.

[249] R. Rajkumar, *Synchronization in real-time systems: a priority inheritance approach*, vol. 151. Springer Science & Business Media, 2012.

[250] X. Wang, H. Sun, T. Deng, and J. Huai, "Consistency or latency? a quantitative analysis of replication systems based on replicated state machines," in *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 1–12, June 2013.

[251] H. Xia, T. Lu, B. Shao, X. Ding, and N. Gu, "Hermes: On collaboration across heterogeneous collaborative editing services in the cloud," in *Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 655–660, May 2014.

[252] H. Suo, Z. Liu, J. Wan, and K. Zhou, "Security and privacy in mobile cloud computing," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 655–659, July 2013.

[253] S. Kudrle, M. Proulx, P. Carrires, and M. Lopez, "Fingerprinting for solving a/v synchronization issues within broadcast environments," *SMPTE Motion Imaging Journal*, vol. 120, pp. 36–46, July 2011.

[254] F. M. F. Wong, C. Joe-Wong, S. Ha, Z. Liu, and M. Chiang, "Mind your own bandwidth: An edge solution to peak-hour broadband congestion," *arXiv preprint arXiv:1312.7844*, 2013.

[255] Juniper Networks, "SRX High Availability Design Guide." `http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/SRX%20High%20Availability%20Deployment%20Guide.pdf`, 2014.

[256] Juniper Networks, "Dynamic Interfaces and Dynamic Subscriber Interfaces Overview." `http://www.juniper.net/techpubs/en_US/junose15.1/topics/concept/dynamic-subscriber-interfaces-overview.html`, 2014.

[257] C. G. Kim and K. J. Kim, "Implementation of a cost-effective home lighting control system on embedded linux with openwrt," *Personal and Ubiquitous Computing*, vol. 18, no. 3, pp. 535–542, 2014.

[258] D. Algorithm. https://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html, 1998.

[259] G. O. system, "Gnutella." http://www.gnu.org/philosophy/gnutella.en.html, 2016.

[260] G. Akhmat, K. Zaman, T. Shukui, and F. Sajjad, "Does energy consumption contribute to climate change? evidence from major regions of the world," *Renewable and Sustainable Energy Reviews*, vol. 36, pp. 123 – 134, 2014.

[261] I. P. on Climate Change (ipcc), "U.s. greenhouse gas inventory report: 1990-2014." œhttp://www.ipcc.ch/report/ar5/wg1/index_fr.shtml, 2016.

[262] I. E. A. (iea), "Energy climate and change - world energy outlook special report." œhttps://www.iea.org/publications/freepublications/publication/WEO2015SpecialReportonEnergyandClimateChange.pdf, 2015.

[263] N. R. D. C. (NRDC), "America's data centers consuming and wasting growing amounts of energy." https://www.nrdc.org/sites/default/files/data-center-efficiency-assessment-IB.pdf, 2014.

[264] O. of Energy Efficiency and R. E. of the government of the US, "10 facts to know about datacenters." http://energy.gov/eere/articles/10-facts-know-about-data-centers, 2014.

[265] A.-C. Orgerie, L. Lefèvre, and I. G. Lassous, "Energy-efficient bandwidth reservation for bulk data transfers in dedicated wired networks," *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1139–1166, 2012.

[266] D. L. Mills and H. Braun, "The nsfnet backbone network," *SIGCOMM Computing Communications Review*, vol. 17, pp. 191–196, Aug. 1987.

[267] R. L. Freeman, *Fundamentals of Telecommunications, 2nd Edition*, ch. 1 INTRODUCTORY CONCEPTS. Wiley, 2005.

[268] "Renater." https://www.renater.fr, 2016.

[269] K. Renard, C. Peri, and J. Clarke, "A performance and scalability evaluation of the ns-3 distributed scheduler," in *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, pp. 378–382, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012.

[270] DELL, "Dell powerconnect 6200 series." http://www.dell.com/downloads/global/products/pwcnt/en/PC_6200Series_proof1.pdf, 2011.

[271] I. Server Technology, "Ug - cwg series pops switched cabinet power distribution unit." https://cdn10.servertech.com/assets/documents/documents/197/original/301-0429-1_Switched_POPS_PDU_Rev_N.pdf?1470331446#_ga=1.88240923.1615207683.1479897126, 2016.

[272] I. Cuadrado-Cordero, A.-C. Orgerie, and C. Morin, "Google doc and picassa use traces." "http://smartnet.gforge.inria.fr", 2014.

[273] "Wireshark." https://www.wireshark.org/, 2016.

[274] E. N. Power, "Energy logic: Calculating and prioritizing your data center it efficiency actions." "http://www.datacentres-ireland.com/files/2016/08/Russell-Bulley-data-center-efficiency_24632.pdf", 2016.

[275] J. G. Koomey, "Growth in data center electricity use 2005 to 2010." "http://www.mediafire.com/file/zzqna34282frr2f/koomeydatacenterelectuse2011finalversion.pdf", 2011.

[276] D. L. Johnson, E. M. Belding, and G. van Stam, "Network traffic locality in a rural african village," in *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, ICTD '12, (New York, NY, USA), pp. 268–277, ACM, 2012.

[277] M. Afanasyev, T. Chen, G. Voelker, and A. Snoeren, "Usage patterns in an urban wifi network," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1359–1372, Oct 2010.

[278] NextDoor, "Nextdoor." https://nextdoor.com/, 2016.

[279] GoNeighbour, "Goneighbour." http://www.goneighbour.org/, 2016.

[280] C. Unnithan, P. S. Ramkumar, A. Babu, S. Joseph, S. Patil, and A. Joshi, "Mobile waste management for smart cities: Monitoring sanitation through living labs," in *Proceedings of International Conference on ICT for Sustainable Development: ICT4SD 2015 Volume 1* (S. C. Satapathy, A. Joshi, N. Modi, and N. Pathak, eds.), (Singapore), pp. 625–633, Springer Singapore, 2016.

[281] P. Neirotti, A. D. Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, "Current trends in smart city initiatives: Some stylised facts," *Cities*, vol. 38, pp. 25 – 36, 2014.

[282] IEEE, "Ieee smart cities initiative." "http://smartcities.ieee.org/", 2016.

[283] IEC, "Orchestrating infrastructure for sustainable Smart Cities," tech. rep., International Electrotechnical Commission, 2015.

[284] IEEE, "Ieee smart cities initiative." "http://smartcities.ieee.org/about.html", 2016.

[285] T. Bakıcı, E. Almirall, and J. Wareham, "A smart city initiative: the case of barcelona," *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 135–148, 2013.

[286] M. Angelidou, "Smart city policies: A spatial approach," *Cities*, vol. 41, Supplement 1, pp. S3 – S11, 2014. Current Research on Cities.

[287] Y. Hu and J. Wang, "Building smart government or developing industry?: Study on the designs of local smart city pilot projects in china," in *Proceedings of the 17th International Digital Government Research Conference on Digital Government Research*, pp. 532–533, ACM, 2016.

[288] F. Delmastro, V. Arnaboldi, and M. Conti, "People-centric computing and communications in smart cities," *IEEE Communications Magazine*, vol. 54, pp. 122–128, July 2016.

[289] Orange, "Orange business: Smart cities." `http://www.orange-business.com/fr/smart-cities`, 2016.

[290] S. Networks, "Global connected & smart city services." `http://www.silverspringnet.com/solutions/smart-cities/smart-cities-city-services/`, 2016.

[291] B. C. Council, "Barcelona ciutat digital." `http://ajuntament.barcelona.cat/estrategiadigital/ca`, 2016.

[292] Cisco, "Cisco enterprise wireless mesh solution overview." `http://www.cisco.com/c/en/us/solutions/collateral/wireless/aironet-1240-ag-series/net_brochure0900aecd806b9180.html`, 2013.

[293] Starbucks, "Starbucks wifi." `https://www.starbucks.com/coffeehouse/wireless-internet`, 2016.

[294] Free, "Freebox." `http://www.free.fr/freebox/index.html`, 2016.

[295] B. C. Council, "Official figures of population in barcelona." `"http://www.bcn.cat/estadistica/angles/dades/barris/tpob/ine/a2015/index.htm"`, 2015.

[296] C. Yu and F. Huan, "Live migration of docker containers through logging and replay," in *Advances in Computer Science Research, International Conference on Mechatronics and Industrial Informatics*, Dec 2015.

[297] D. Serrano, S. Bouchenak, Y. Kouki, F. A. de Oliveira Jr., T. Ledoux, J. Lejeune, J. Sopena, L. Arantes, and P. Sens, "{SLA} guarantees for cloud services," *Future Generation Computer Systems*, vol. 54, pp. 233 – 246, 2016.

[298] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing Amazon EC2 Spot Instance Pricing," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 304–311, 2011.

[299] S. A. Noor, R. Hasan, and M. M. Haque, "CellCloud: A Novel Cost Effective Formation of Mobile Cloud Based on Bidding Incentives," in *IEEE International Conference on Cloud Computing (CLOUD)*, pp. 200–207, 2014.

[300] A. L. Jin, W. Song, and W. Zhuang, "Auction-Based Resource Allocation for Sharing Cloudlets in Mobile Cloud Computing," *IEEE Transactions on Emerging Topics in Computing*, 2015.

[301] H. Wang, H. Tianfield, and Q. Mair, "Auction Based Resource Allocation in Cloud Computing," *Multiagent Grid Systems*, vol. 10, pp. 51–66, Jan. 2014.

[302] P. Samimi, Y. Teimouri, and M. Mukhtar, "A combinatorial double auction resource allocation model in cloud computing," *Information Sciences*, vol. 357, pp. 201 – 216, 2016.

[303] K. Xu, Y. Zhang, X. Shi, H. Wang, Y. Wang, and M. Shen, "Online combinatorial double auction for mobile cloud computing markets," in *2014 IEEE International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, Dec 2014.

[304] Z. Sun, Z. Zhu, L. Chen, H. Xu, and L. Huang, "A combinatorial double auction mechanism for cloud resource group-buying," in *IEEE International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, 2014.

[305] I. T. Union, "Ict facts and figures." `https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf`, 2013.

[306] Facebook, "Internet.org." `https://info.internet.org/en/approach/`, 2016.

[307] K. Chung and R. C. Park, "P2p cloud network services for iot based disaster situations information," *Peer-to-Peer Networking and Applications*, vol. 9, no. 3, pp. 566–577, 2016.

[308] A. S. Elmaghraby and M. M. Losavio, "Cyber security challenges in smart cities: Safety, security and privacy," *Journal of advanced research*, vol. 5, no. 4, pp. 491–497, 2014.

Résumé

1 Contexte et motivation

Le Cloud Computing est une technologie qui mutualise des ressources de calcul, de stockage et de communication pour les offrir aux clients en s'appuyant sur un modèle économique de facturation à la demande. Il fournit ainsi des services à la demande, flexibles et adaptés aux besoins des clients ; rendant ainsi cette technologie accessible à un large éventail d'utilisateurs qui ne peuvent pas se permettre de gérer une puissance de calcul importante, tel un centre de données privé. Le Cloud Computing sous-tend de plus en plus de services Internet car la mutualisation des ressources permet de fournir des services plus efficaces que des installations privées localisées. En bénéficiant des économies d'échelle, les infrastructures distribuées de Cloud peuvent gérer efficacement leurs ressources et offrir des capacités de stockage et de calcul quasiment illimitées, tout en minimisant les coûts pour les utilisateurs.

1.1 Consommation énergétique du Cloud

Pour offrir leurs services, les fournisseurs de services Cloud construisent des centres de données où sont installées les ressources de calcul et de stockage et auxquels les clients se connectent. Bien que cette architecture se soit avérée efficace du point de vue des utilisateurs, les services Internet continuent de représenter une part importante de la consommation mondiale d'énergie. La consommation d'énergie actuelle des centres de données et des réseaux de cœur (réseaux principaux connectant les points de présence des opérateurs Internet) est estimée à 2 à 4 % de la production mondiale d'énergie.

L'utilisation d'énergie par les technologies Internet devrait augmenter dans les prochaines années du fait de l'augmentation des personnes connectées. De plus, avec l'apparition de l'informatique mobile et des objets connectés, le nombre d'équipements reliés à Internet connaît lui aussi une forte augmentation, en plus d'une extension de leur répartition. Cette utilisation croissante d'Internet impose de nouvelles limites aux appareils mobiles en termes de ressources et de connectivité. Le Cloud doit ainsi faire face à de nouveaux défis en matière de consommation énergétique et de qualité de service.

La croissance prévue, tant pour le nombre de nouveaux utilisateurs que pour leur répartition, a un impact non seulement sur l'environnement, mais aussi sur les sociétés qui gèrent les services Internet. D'une part, la multiplication des nouveaux appareils connectés oblige le réseau des fournisseurs de services Internet à accroître leurs ressources, lesquelles sont limitées aux matières brutes et à l'espace disponibles dans les installations ; et cela augmente la consommation d'énergie des réseaux de télécommunication. D'autre part, les calculs qui peuvent être pris en charge par un centre de données sont limités aux ressources qu'il peut accueillir. En conséquence, le nombre de centres de données dans le monde continue de croître. Comme pour les fournisseurs d'accès à Internet (FAI), cette situation a une incidence à la fois sur la facture d'électricité de ces sociétés qui gèrent des Clouds et sur leur impact environnemental, imputable à l'augmentation de leur consommation d'énergie.

1.2 Qualité du service dans le Cloud

Avec l'évolution des appareils connectés, leurs besoins ne cessent de croître. Par exemple, la qualité de service attendue pour un ordinateur personnel utilisé pour des jeux vidéo en ligne est nettement supérieure à la qualité de service attendue d'un serveur de messagerie déployé il y a 10 ans. La qualité de service fournie par un service Internet est aujourd'hui un problème pour de nombreux appareils connectés. L'un des principaux facteurs qui influent sur cette qualité de service est la distance que les données doivent parcourir. En réduisant la distance entre les clients et les centres de données, la latence des services est diminuée, améliorant ainsi la qualité de service. Cela touche en particulier les applications sensibles à la latence. Le retard dans les communications provoqué par l'utilisation d'architectures axées sur des centres de données centralisés et donc distants peut ne pas être acceptable pour certaines applications.

Une telle situation oblige les fournisseurs de services, tels Google ou Amazon, à changer leur conception du Cloud, en répartissant leurs ressources entre différents emplacements, afin d'offrir un service plus proche que des services centralisés dans un seul centre de données. Le recours à plusieurs centres de données est une approche préférable à celle consistant à en utiliser un seul de grande taille, regroupant les services qui sont destinés à une région particulière telle que l'Europe. Une approche basée sur plusieurs centres de données réduit également la distance entre les services et les utilisateurs, et fait un meilleur usage de l'énergie. En effet, les utilisateurs qui font appel à ce service partagent normalement la même plage horaire, et ainsi, pendant la nuit, certains équipements peuvent être arrêtés pour économiser de l'énergie. Cette situation est de plus en plus fréquente avec la multiplication des initiatives de ville intelligente dans le monde entier (Santa Cruz, Amsterdam, Barcelone, etc.), où les autorités locales déploient des plateformes d'équipements interconnectées pour fournir des informations utiles à leurs citoyens via des réseaux sans fil. Ces informations peuvent être utilisées dans des services tels que la réduction du trafic ou la gestion des situations d'urgence. Cette approche décentralisée reste cependant inefficace en termes d'utilisation de l'énergie et de qualité de service pour les applications où la propagation des données est limitée à une zone géographique davantage localisée que la zone ciblée par le centre de données, qui couvre généralement un ou plusieurs pays.

2 Problématique

Nous visons ici à réduire la consommation énergétique du Cloud, tout en fournissant une meilleure qualité de service à l'utilisateur. Pour cela, nous exploitons la faible propagation des données de certaines applications. Comme cela a été étudié précédemment dans la littérature, dans de nombreux cas, les données se propagent relativement peu par rapport à leur source. En d'autres termes, plusieurs utilisateurs se servant d'une application se trouvent géographiquement proches les uns des autres. Bien qu'il existe une multitude d'applications utilisant le Cloud, nous avons décidé de nous concentrer sur les applications collaboratives qui fournissent un accès simultané à plusieurs utilisateurs. Cela implique que tous les utilisateurs doivent travailler sur les mêmes données, avec une réduction du délai entre le moment où un utilisateur met à jour les données dans l'application et celui où les autres reçoivent cette mise à jour. La latence est par conséquent importante dans ce type d'applications. De plus, nous souhaitons offrir une qualité de service acceptable aux utilisateurs tout en garantissant une faible consommation énergétique.

Ce problème n'est pas résolu par les approches distribuées existantes. Dans les approches de la littérature, le calcul est exécuté là où les données sont stockées, c'est-à-dire sur plusieurs appareils différents. Bien que les approches distribuées offrent une plus grande résilience aux défaillances et une distribution importante des données sur un réseau, elles ont certains défauts qui les rendent inaptes à résoudre le problème abordé dans ce travail. En effet, par nature, les solutions distribuées sont dépourvues d'une gestion centralisée. Même si l'absence de centralisation accélère le processus, les solutions distribuées souffrent d'une réplication excessive des données par rapport aux solutions centralisées. La forte réplication des données et du calcul des approches distribuées les rendent inappropriées pour les applications interactives avec plusieurs contributeurs, car la réplication peut provoquer des conflits entre les différentes versions. De plus, cette réplication augmente l'utilisation de la bande passante du réseau en raison des synchronisations fréquentes entre les versions des données hébergées en différents endroits et des protocoles de découverte des contributeurs. En conséquence, la consommation d'énergie augmente également dans les réseaux des fournisseurs d'accès à Internet (FAI), étant donné que le trafic réseau est plus important et nécessite donc plus de ressources réseau.

3 Notre approche

Le défi relevé dans cette thèse consiste à concevoir une solution Cloud efficace en énergie qui offre une meilleure qualité de service que l'état de l'art, sans dépendre d'une infrastructure particulière. D'un côté, on voit qu'une approche qui centralise le calcul dans des grands centres de données devient contre-productive du point de vue de l'efficacité énergétique. De l'autre côté, le recours aux approches distribuées de type pair-à-pair devient inefficace du point de vue de l'opérateur réseau. De plus, l'impact des réseaux des fournisseurs d'accès à Internet (FAI) qui interconnectent l'utilisateur final et les centres de données a été négligé dans la littérature. Nous proposons une approche qui vise à la décentralisation des ressources, que nous appelons microclouds, et qui intègre les réseaux de télécommunication comme éléments participant à l'infrastructure du cloud. Ainsi, un microcloud est formé par un ensemble d'appareils (appareils clients et équipements réseau) qui utilisent un même service.

Pour gérer des microclouds du point de vue plateforme, nous propose *GRaNADA* (GReen and Network-Aware Decentralized Architecture), un système *Platform-as-a-Service* décentralisé qui distribue géographiquement le calcul entre les appareils impliqués dans la communication (tels que des clients et des périphériques réseau) pour économiser l'énergie et fournir une meilleure qualité de service. *GRaNADA* n'élimine pas le besoin de centres de données centralisés, mais leur retire tout les calculs et données susceptibles d'être localisés, afin qu'ils soient traités au plus près des utilisateurs finaux du service. Parallèlement à *GRaNADA*, nous proposons *DEEPACC*, un protocole de routage pour les microclouds, qui planifie une distribution des connexions entre les nœuds basée sur des critères d'efficacité énergétique et de qualité de service du système. Avec cette solution, les services sont hébergés sur les appareils concernés (les appareils clients, ou équipements réseau). L'ensemble des appareils concernés forme une plateforme d'overlay qui est appelée *microcloud*.

4 Validation

Nous avons décidé d'appliquer GRaNADA dans deux contextes pratiques différents afin d'évaluer sa pertinence : dans un réseau central et dans une infrastructure de ville intelligente. Dans le premier cas, l'adaptation de GRaNADA à un réseau central vise à montrer comment l'utilisation des microclouds permet d'économiser de l'énergie. Dans le second cas, l'adaptation de GRaNADA à une infrastructure de ville intelligente vise à démontrer comment l'emploi de microclouds fournit une meilleure qualité de service au client.

4.1 Scénario de réseau central

Nous nous concentrons sur la réduction de la consommation énergétique des Clouds en accordant une attention particulière au réseau, qui a été négligé dans les travaux précédents de la littérature. Le premier scénario d'étude adapte GRaNADA pour son déploiement dans un système à grande échelle tel qu'un réseau central. Pour évaluer notre système, nous avons construit un prototype de GRaNADA et l'avons simulé sur différents réseaux à l'aide de ns3, un simulateur réseau niveau paquet très utilisé dans la communauté. Nous avons simulé un réseau central et comparé notre solution à une approche complètement décentralisée et à une approche centralisée. Les résultats montrent que notre solution de microcloud permet d'économiser jusqu'à 75 % de l'énergie par rapport à une approche décentralisée, et consomme 10 fois moins que l'architecture centralisée classique.

4.2 Scénario des villes intelligentes

Nous avons également appliqué notre solution dans un contexte plus localisé pour étudier ses performances en terme de qualité de service. Dans ce contexte, nous laissons entièrement de côté les centres de données et nous localisons toute l'infrastructure du microcloud dans une zone densément peuplée, mais de petite dimension (par exemple un quartier). Pour prouver la validité de notre système, nous avons évalué la pertinence de GRaNADA dans le contexte des villes intelligentes. Nous avons utilisé un scénario qui considérait une plateforme basée sur des services de proximité, c'est-à-dire des services fournis aux utilisateurs situés dans une zone de voisinage, concernant, par exemple, des travaux routiers prévus, des réunions communales, des boutiques locales, etc. Via l'utilisation de GRaNADA, nous proposons une plateforme pour ces services de la vie réelle qui améliore le paradigme de ville intelligente et que nous avons intitulée *microcities*. Dans ce scénario, nous tirons parti des ressources inutilisées des appareils des utilisateurs et des infrastructures de la ville intelligente afin d'héberger des services conçus pour un quartier particulier. Pour évaluer notre système, nous avons adapté le prototype existant de GRaNADA et l'avons simulé sur différents réseaux à l'aide de ns3. Les simulations montrent qu'avec la restriction des calculs à une zone géographique restreinte via l'utilisation de *microclouds*, l'utilisateur bénéficie d'une qualité de service nettement supérieure à celle offerte par des Clouds centralisés dans un centre de données.

5 Favoriser l'acceptation de notre solution

Les utilisateurs peuvent être réticents à adopter une nouvelle technologie où certains d'entre eux contribuent davantage à fournir des ressources que les autres (et certains pas du tout). Dans la littérature, on parle de problème du passager clandestin. Ce problème se pose lorsque plusieurs utilisateurs utilisent les services offerts par le système sans contribuer pour leur part. Pour résoudre ce problème et stimuler l'adoption de notre technologie, nous proposons la création d'un marché de services entre les utilisateurs, notre dernière contribution consistant ainsi à assurer la rémunération des utilisateurs pour l'utilisation de leurs ressources. Nous proposons un système d'enchères adapté au paradigme de *microcities*, que nous avons appelé *micromarkets*. Nous montrons qu'en utilisant ce système, les utilisateurs reçoivent un service mieux adapté à leurs besoins, ce qui améliore leur degré de satisfaction (différence entre la qualité de service demandée et la qualité de service reçue). Par ailleurs, les utilisateurs qui louent leurs ressources reçoivent une compensation pour leurs services, tandis que les fournisseurs du service attirent davantage d'utilisateurs.

6 Organisation du manuscrit

Ce manuscrit est structuré comme suit. Dans le chapitre 2, nous présentons une évaluation des différentes solutions existantes et un état de l'art des approches concurrentes. Dans le chapitre 3, nous décrivons le concept de *microcloud* et un prototype de l'architecture logicielle associée (*GRaNADA* et *DEEPACC*). Dans le chapitre 4, nous appliquons GRaNADA dans les réseaux de cœur et nous évaluons sa pertinence, en nous focalisant sur sa consommation d'énergie, par rapport à des approches centralisées et décentralisées. Dans le chapitre 5, nous appliquons GRaNADA dans le contexte d'une ville intelligente où les utilisateurs se trouvent dans des zones géographiques localisées et nous évaluons sa pertinence du point de vue de la qualité du service. Dans le chapitre 6, nous proposons un modèle économique basé sur des enchères doubles pour favoriser l'utilisation des microclouds. Enfin, dans le chapitre 7, nous concluons cette thèse et nous proposons des pistes pour des recherches futures.