Computer Networks 93 (2015) 531–542

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Managing mobile cloud computing considering objective and subjective perspectives

Marcelo Antonio Marotta [a], Leonardo Roveda Faganello [a], Matias Artur Klafke Schimuneck [a], Lisandro Zambenedetti Granville [a], Juergen Rochol [a], Cristiano Bonato Both [b,*]

[a] Institute of Informatics – Federal University of Rio Grande do Sul Av. Bento Gonçalves, 9500 – Porto Alegre, RS - Brazil
[b] Federal University of Health Sciences of Porto Alegre, Brazil Av. Independência, 2293 – Santa Cruz do Sul - RS / Brasil

ARTICLE INFO

Article history:
Received 16 March 2015
Revised 10 August 2015
Accepted 19 September 2015
Available online 24 October 2015

Keywords:
Mobile cloud computing
Network management
QoE
QoS

ABSTRACT

Mobile Cloud Computing enables mobile devices to augment constrained resources such as processing, storage, and battery autonomy by using the cloud infrastructure. As the network is a key element in integrating mobile devices to the cloud, a proper management of the mobile cloud computing environment is necessary. Such a management must take into account two main perspectives: administrator's and end-user's perspectives. The administrator is usually concerned with a more objective perspective based on Quality of Service parameters, such as throughput, delay, and jitter. On the other hand, the end-user has a more subjective perspective, observing his/her Quality of Experience when using a mobile cloud application or service. In this article, we introduce a management model and architecture for mobile cloud computing, exploiting both objective and subjective perspectives. As a proof of concept, we prototyped the architecture in a management system called CoLisEU, which allowed us to investigate this architecture and we discuss the benefits of the combined objective and subjective perspectives in our management architecture.

1. Introduction

Mobile Cloud Computing (MCC) is the paradigm that integrates mobile devices and cloud computing. MCC allows the augmentation of mobile devices' constrained resources, such as processing, storage, and battery autonomy by using cloud services [1]. Counting on augmented resources, mobile devices can execute more sophisticated versions of key applications and services, such as mobile learning, e-commerce, and healthcare [2]. This is possible because tasks in MCC are partially performed in mobile devices and partially computed in the cloud. The network that interconnects mobile devices and the cloud largely impacts the proper execution of MCC applications and services. Besides, this network also allows MCC to perform a second augmentation, now in terms of coverage, extending the cloud towards the edge of the network, where the mobile devices and End-users lie [3]. In this article, we address the important issue of managing the MCC environment, focusing on the networking aspects. Such a management must take into account two main perspectives: the network Administrator's perspective and the End-user's perspective.

The Administrator is usually concerned with Quality of Service (QoS) parameters (*e.g.*, throughput, delay, and jitter [4]), which provide an *objective* view about the quality of the network. By observing QoS-related measurements, the Administrator can tune the operation of underlying

* Corresponding author. Tel.: +55 5193080875.
E-mail addresses: mamarotta@inf.ufrgs.br (M.A. Marotta), lrfaganello@inf.ufrgs.br (L.R. Faganello), makschimuneck@inf.ufrgs.br (M.A.K. Schimuneck), granville@inf.ufrgs.br (L.Z. Granville), juergen@inf.ufrgs.br (J. Rochol), cbboth@ufcspa.edu.br, cbboth@inf.ufrgs.br (C.B. Both).

http://dx.doi.org/10.1016/j.comnet.2015.09.040
1389-1286/

infrastructures, including wireless infrastructure, backhaul, backbone, provider infrastructure, and cloud infrastructure [1]. The End-user, in turn, tends to neglect QoS parameters and focus instead on his/her experience when using an MCC application or service. This experience can be measured by observing Quality of Experience (QoE) parameters, such as satisfaction level with application navigation, response time, and cloud ubiquity [5]. Since each End-user has a personal experience, QoE parameters provide a *subjective* view about MCC applications and services. Although different to the objective view obtained from QoS observation, the subjective, QoE-based observation of MCC is also important because it is directly related to the applications and services consumed by the End-user.

The combined observation of QoS and QoE parameters are fundamental to the management of MCC. However, such management is still underexploited by the research community. Some preliminary investigations point out that QoS and QoE can serve as guiding paradigms [6,7] for improving MCC management [8]. Other investigations focus on proposing architectural models for MCC management that are centered on the End-user [9,10] or on cloud applications [2], but neglect QoE parameters. Therefore, there is a need for concrete architectural models that explicitly take into account both QoS and QoE together.

In this article, we investigate the management of MCC through an architecture that considers both objective (QoS-based) and subjective (QoE-based) perspectives. In Section 2, we review MCC basic concepts, as well as the state-of-the-art on MCC management. Afterwards, we highlight five key requirements for managing the MCC environment. Next, a mapping of traditional management entities (*i.e.,* agent, gateway, and management application) to the MCC environment is introduced. In Section 3, we describe a management architecture that explicitly takes into account both QoS and QoE. As a proof of concept, in Section 4, we prototyped the proposed architecture in a management system called CoLisEU,[1] whose details are presented in the article as well. To evaluate our prototype, in Section 5, we analyze one thousand samples provided by End-users about their satisfaction using a wireless infrastructure to request cloud services. Finally, the benefits of the combined analysis of QoS and QoE in our MCC management architecture are summarized in the concluding section.

2. Mobile Cloud Computing

In this section, we present the components of an MCC infrastructure. Next, we present the state-of-the-art of the MCC management. Afterwards, we detail key management requirements considering the MCC environment.

2.1. MCC infrastructure

MCC enables the augmentation of computing resources of mobile devices in an environment composed of the following infrastructures: (*i*) *wireless infrastructure*, (*ii*) *backhaul*, (*iii*) *backbone*, (*iv*) *provider infrastructure*, and (*v*) *cloud infrastructure*. These components are depicted in Fig. 1. The *End-user*, through his/her *Mobile Device* establishes communication with a *Base Station* or an *Access Point*, requesting a resource (*e.g.,* storage or processing) augmentation from the cloud. This request may be based on different wireless technologies, such as LTE or WiFi. After reaching the *Base Station/Access Point*, the request is forwarded through the *Backhaul* to an Internet Service Provider (ISP).

The *Backbone* routes the request along different ISPs. When the request reaches the destination ISP, it accesses the *Provider Infrastructure*. This infrastructure is similar to the *Backhaul*, only with typically larger capacity links. The target Cloud then receives the request and allocates resources inside the *Cloud Infrastructure*. In the Cloud, virtual nodes communicate with one another through virtual links. These links are created by the Cloud administrators as an abstraction of the real network links with specific characteristics, such as capacity, routing protocol, and virtual node endpoints. Finally, the Cloud provides the requested resources, replying to the *Mobile Device* across the five infrastructures.

Each infrastructure presents different problems, such as intermittent wireless signal, insufficient coverage areas, traffic overhead, non-optimal service level agreements, and virtual link misconfiguration [11]. These problems may degrade both QoS and QoE. To try to avoid this degradation, MCC management must monitor each infrastructure to detect these problems. When problems are detected, MCC management reconfigures and tunes the internal components of the five infrastructures, in order to lead the network to a state that again satisfies the *End-user* expectations, thus keeping QoE at satisfactory levels.

2.2. MCC management

MCC extends from the *End-user* to the *Cloud Infrastructure*. Therefore it is important that the MCC management strategy takes into account the perspectives of both the *Administrator* and *End-user*. The *Administrator* has a more objective view of the managed network, typically observing QoS parameters to detect specific problems, such as link bottlenecks or network node inactivity [4]. However, solely relying on QoS parameters may cause Administrators to be unaware of whether the *End-user* expectations are fulfilled (*i.e.,* QoE). As opposed to Administrators, *End-users* have a more subjective perspective of the network, based on QoE. As such, aspects that escape the Administrator's objective analysis (*e.g.,* satisfaction level, network efficiency, and Cloud ubiquity), end up being confined to the End-user's point-of-view. That, however, is inadequate and QoE must be observed too. On the opposite side, QoE evaluation alone is not able to accurately point out specific problems in any of the five MCC infrastructures. We thus argue that MCC management must be based on a joint observation of both QoS and QoE.

In reviewing the state-of-the-art on the management of MCC, we divide the literature work into two groups: (*i*) works that combine the use of QoS and QoE and (*ii*) works that focus on architectural models for MCC management. The investigations from the first group indicate that the traditional QoS-based management alone is insufficient to guarantee *End-user's* satisfaction [6–8,12]. Meanwhile, management architectures (from the second group) do not consider QoE, as described in the surveys of Rahimi, Ren, and Liu [2], and

[1] CoLisEU: https://coliseu.rnp.br .

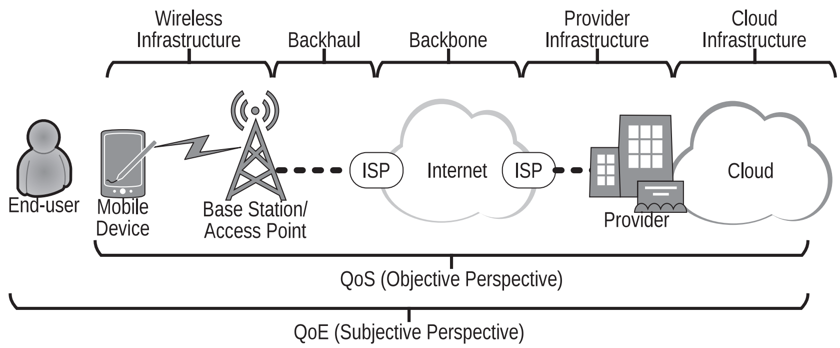


Fig. 1. Mobile cloud computing environment.

Abolfazli *et al.,* [1]. Although most current MCC management architectures represent new proposals, they are not based on the quite large and solid management principles established by the network and services management research field. For example, management entities, such as agents and management stations, are weakly discussed and presented, remaining unmapped in today's MCC management environments.

In the network management research field, two management entities are well known and explored: (*i*) agents and (*ii*) managers. The agent is a software module placed inside an infrastructure component and is responsible for monitoring and controlling local parameters, such as available memory and Maximum Transmission Unit (MTU). The manager, in turn, is a role assumed by a network node, *e.g.,* computers, set-top-boxes, or routers, and is responsible for retrieving information from agents and managing a network slice or domain. The main difference between other investigations on the management of MCC and our research is that we provide a mapping of the management entities to the MCC environment, considering the joint observation of QoS and QoE. In this context, we summarize the main management requirements considering the five infrastructures of an MCC environment and the observation of QoS and QoE in the Table 1.

In Table 1, the five functional areas Fault, Configuration, Accounting, Performance, and Security (FCAPS) [13] are mapped into key requirements for any MCC management systems. The MCC requirements are not limited to the presented Table 1, for example, exploring the heterogeneity or scalability of the MCC environment would lead to other requirements. However, FCAPS requirements are broader, being able to include other requirements.

3. MCC management architecture

In this section, we first show an MCC environment in terms of management entities. Next, we propose an architecture for the MCC management system, considering the perspectives of End-users and Administrators. In addition, we detail how the requirements listed in the previous section are addressed.

As can be seen in Fig. 2, *Agent* is an MCC application that can be remotely programmed or triggered by other applications within mobile devices to measure End-users' QoE and network QoS. *Gateways* are intermediary nodes installed at the edge of the network that bridge communications between *Management Applications* and *Agents*. *Management Application* corresponds to software installed in a cloud virtual node, handled by the cloud infrastructure, to provide management tools for Administrators and End-users. Moreover, cloud components, such as the *AAA Server*, as well as other *Cloud Resources*, are presented inside the cloud. *Base Stations* and *Access Points* are shown as intermediate equipments between *Mobile Devices* and logical links to *Gateways*. Finally, the Administrator and End-users that participate in the MCC environment are also shown using a desktop and mobile devices, respectively.

In such an environment, we propose a management architecture where *Agent*, *Gateway*, and *Management Application* are placed inside different MCC components, respectively: *mobile device*, *virtual network node*, and *virtual cloud node*. These architecture components are shown in Fig. 3, and according to each management entity, are better described in the next subsections.

3.1. Agent

The MCC infrastructure must be properly managed through the monitoring of its evaluation status, which is composed of several QoS parameters (*e.g.,* throughput, *Mobile Device* Received Signal Strength Indicator - RSSI, and Round Trip Time - RTT), as well as QoE measurements about the MCC infrastructure. Currently, the cloud, the provider, and the core infrastructures can have their state partially exposed with QoS measurements using, for example, traditional SNMP or NETCONF agents. In turn, the backhaul and the wireless infrastructures – where Administrators have loose/partial control and cannot freely deploy agents – have their states defined based on some statistical estimate of link usage, *e.g.,* bit rate of an entire domain. Whereas, the gathering of QoE measurements for each of the five infrastructures is a metric that is difficult to obtain, mainly because End-users have indirect contact with most of these infrastructures and are unable to give a proper review of a particular one. It means that QoE and QoS measurements are gathered partially preventing the retrieving of a complete evaluation status of the MCC environment. In this sense, the cooperation of End-users that can use their *Mobile Devices* to gather QoS and QoE measurements involving MCC infrastructures becomes fundamental.

Table 1
MCC management requirements.

Requirement	MCC environment
Fault [14]	The management system must be aware of the five infrastructure faults, *e.g.,* nodes failures, to avoid QoS and QoE degradation.
Configuration [15]	The management system must reconfigure the five infrastructures to achieve correctness and autonomy based on QoS and QoE.
Accounting [16]	The management system must monitor and measure the usage of the five infrastructure through QoS and QoE for billing correctness and auditing purposes.
Performance [17]	The management system must support a large number of mobile devices performing asynchronous communications to avoid compromising the proper operation of the MCC environment.
Security [18]	The management system must Authenticate, Authorize, and Account (AAA), *End-users* actions inside the MCC environment, avoiding impersonation attacks as well as providing a stronger auditing.

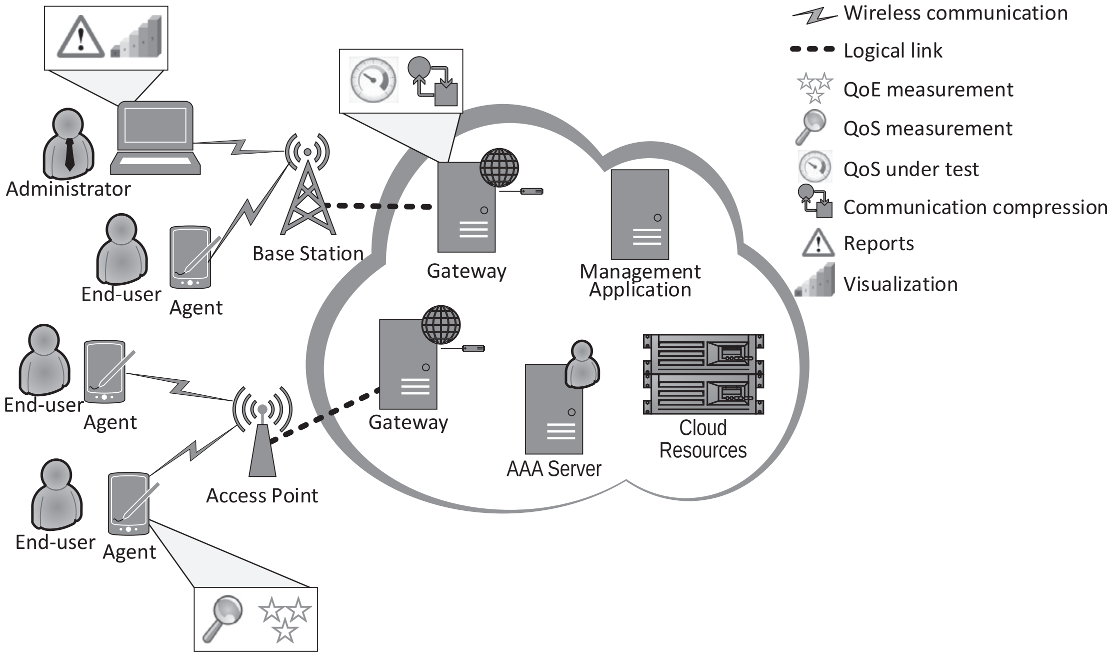


Fig. 2. MCC management environment.

To allow End-users to participate actively in the management of MCC environments, a new *Agent* placed inside the *Mobile Device* is proposed. The *Agent* is responsible for gathering QoS measurements from the wireless and backhaul infrastructures. In addition, the *Agent* provides capabilities for End-users to report their QoE about the entire MCC infrastructure. *Agent* entities must cope with the specific hardware and software characteristics of a *Mobile Device*, such as *Wireless Network Interfaces*, *e.g.,* WiFi or LTE cards, and *Operating Systems* (OS), *e.g.,* Android, iOS, or Windows Phone, since the performance of QoS and QoE gathering is largely influenced by these characteristics.

Inside the *Agent* we propose a main module called *Agent Controller* that communicates with four lower level modules, *QoS Meter*, *System Extractor*, *AAA client*, and *Service Interface*, as well as four upper level modules, *QoE Meter*, *Graphic Viewer*, *Cache*, and *MCC Applications*, described below.

- *QoS Meter* is responsible for collecting QoS measurements from the wireless and backhaul infrastructures of the MCC environment. To account for different QoS metrics, the *QoS Meter* must use a set of well established network measurement tools, such as perfSonar from Internet2. In terms of wireless infrastructure metrics, the *QoS Meter* must account for downlink throughput, delay, and RSSI. Whereas, for the backhaul infrastructure, the *Gateway* must be used with the *Agent* to gather other metrics, such as RTT, jitter, and uplink throughput. It is important to notice that the *QoS meter* only accounts for backhaul and wireless infrastructure, whereas traditional SNMP

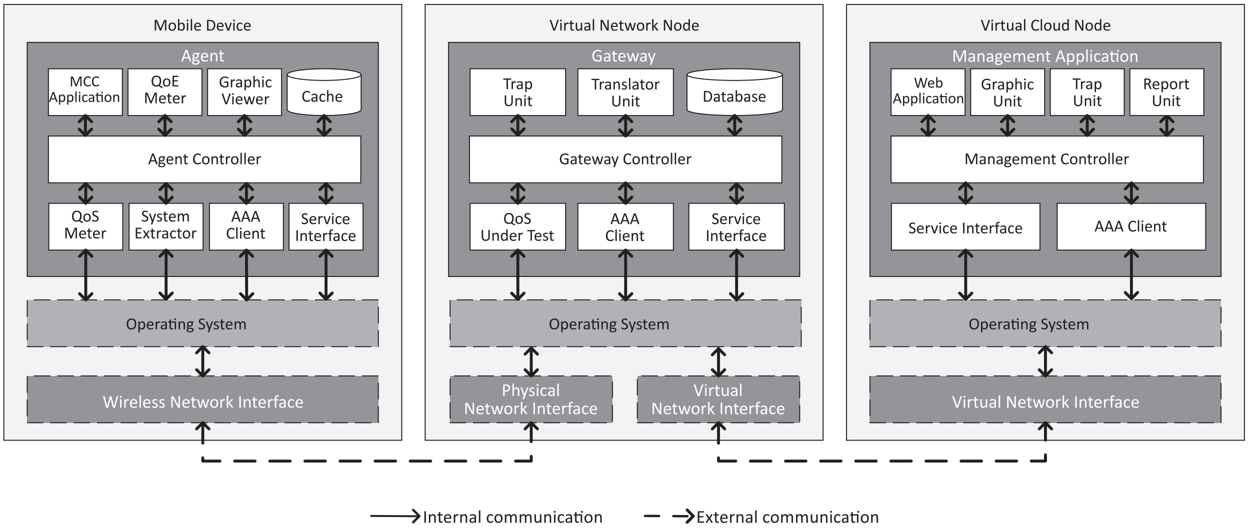


Fig. 3. MCC management architecture.

and NETCONF agents are used for gathering QoS metrics in the core, provider, and cloud infrastructures.

- *System Extractor* retrieves meaningful information about the software and hardware details of the *Mobile Device*. Examples of gathered information are the OS branch and version, device manufacturer, and battery level. In addition, the system extractor is responsible for control system parameters, for example, IP Maximum Transmission Unit (MTU) and TCP Maximum Segment Size (MSS).
- *AAA client* accounts for the authentication of mobile devices, authorizing other modules to perform their actions, and audit each action performed by the *Agent*. This means that each measurement gathering will be secured according to the AAA solution employed, for example, OAuth or RADIUS.
- *Service Interface* homogenizes the communication among *Agents* and *Gateways*, providing a set of services that can be remotely consumed by *Gateways*. In addition, this module defines how each other modules can consume different services provided by *Gateways*, *e.g.,* store QoE measurement or store QoS measurement.
- *QoE Meter* is proposed to be used by other *MCC Applications* to gather QoE from End-users as a default tool to *Mobile Devices*. It means that the MCC management can receive QoE measurements in a homogeneous manner that measures the user satisfaction accounting for different types of *MCC Applications*. In addition, *QoE Meter* provides graphical input tools to interact with End-users to gather their experience about the MCC environment.
- *Graphic Viewer* enables different *MCC Applications* to retrieve chart visualizations of QoS and QoE measurements gathered from the End-User mobile device. These visualizations can be later displayed to the End-users according to the *MCC Application* usage.
- The *Agent Controller* is responsible for managing the other modules' processing requests from the *Virtual Cloud Node* or providing communication among higher and lower level modules.
- *Cache* allows offloading operations, *i.e.,* it stores temporary offline information to be used later [19]. These offloading operations are particularly important for MCC infrastructures, mainly when End-users are facing disconnections and for compressing information before sending to *Gateways*.
- *MCC Applications* are mobile applications that can take advantage of all the other modules from the *Agent* to perform gathering of QoE and QoS. These applications can also use underlying functions for standardization purposes. For example, *MCC Applications* can use the *Agent* as a platform for AAA as much as for the retrieving of visualizations of the network state.

Considering the design of the *Agent* component, we can improve the *Accounting* requirement by monitoring End-users' usage of MCC, such as described in Section 2.2. In addition, the *Security* requirement is provided by the *AAA client*, using the *Mobile as a Representer* approach, where each End-user can be represented by a virtualized entity in the cloud through his/her mobile device [10]. The *Fault* requirement is met using the System Extractor to send reports about malfunctions as well as QoS and QoE degradation to be analyzed by the *Gateways*. *Configuration* is assured by the Service Interface by standardizing the communication protocol among components, as well as the messages format.

3.2. Gateway

To intermediate the communication between mobile devices and the cloud environment, *Gateways* placed inside *Virtual Network Nodes* between the wireless and cloud infrastructures are used. These gateways can temporarily store measured data from *Agents* about the MCC environment. Periodically, the stored data is compressed and sent to a centralized server in the cloud infrastructure. This approach avoids flooding the network with several asynchronous messages, greatly improving the scalability of the management system. *Gateways* also can assume the role of intermediary managers, executing tasks, such as performing polling on *Mobile Devices* or gathering information from the cloud, provider, and core infrastructure through traditional agents. *Gateways* aim to shorten the time taken to exchange management messages between *Agents* and the cloud infrastructure.

Similar to the *Agent*, the *Gateway* must be designed considering that a *Virtual Network Node* may present different OS and Network Interfaces. However, these interfaces are classified in Physical and Virtual. The first is connected to the backbone. The second is connected to an internal network within the core or the cloud infrastructure. Inside the *Gateway*, we specify seven modules, as below:

- *QoS Under Test* is responsible for answering the requests made by the *QoS Meter* in the *Agent*, composing a Client-Server paradigm. In this sense, the *Gateway* can be considered as a measurement point, where *Agents* can measure their network through different tools, *e.g.,* iPerf, OWAMP, or OWPING.
- *AAA client* provides a bridge service for *Mobile Devices* to perform AAA through the *AAA Server*, as depicted in Fig. 2. Authentication requests allow *Gateways* to secure the information provided by each *Agent* by associating it to a trusted and unique ID. It is important to notice that centralized AAA solutions, such as RADIUS, can also be used, excluding *Gateways* from the AAA process. However, a distributed AAA approach favors security and scalability. It means that *Gateways* can distribute content inside the MCC environment to End-users taking advantage of distributed content network concepts in a secure manner.
- *Service Interface*, in the same manner as in the *Agent*, standardizes the communication among *Agent-Gateway* and *Gateway-Manager*. In contrast to *Agents*, the *Gateway Service Interface* module communicates with the *Translator Unit* for compressing and decompressing of messages.
- *Translator Unit* performs all the compressing, decompressing, and translation of messages. Before being sent, any information must be grouped and processed by the *Translator Unit*. This module defines which information will be compressed to be sent to the *Management application*. In addition, received compressed information must be decompressed by the *Translation Unit* using a previous established encryption key for security purposes. Finally, the *Translation Unit* module may also organize information in different formats, *e.g.,* JavaScript Object Notation (JSON) to eXtensible Markup Language (XML).
- A *Database* stores temporary information received from *Agents*, to be later processed by other modules and forwarded to the *Virtual Cloud Node*. Also, the *Gateway* database contains stored actions of each *Agent* for auditing purposes.
- *Trap Unit* handles the notification of undesired events that may occur within the MCC infrastructures, such as signal interference, low throughput, gateway redirections due to a node inactivity, among others. Since all these events are usually unexpected, the *Trap Unit* of each *Gateway* must be always active to receive trap messages. As soon as a trap occurs, depending on its severity, the *Gateway* should perform at least one of three actions: (*i*) forward the trap immediately to an *Alert Unit* on a *Management Application*, (*ii*) cache the trap to be sent later, or (*iii*) perform an extra task. In the first action, the *Gateway* sends the trap message containing a severity level indicator, informing that a drastic event occurred. In the second, a minor severity problem is cached to be sent later. In the third, the *Gateway* may perform any other task, according to the architecture implementation. For example, the *Gateway* may be programmed to execute a custom script in the case of a network interface going down.
- *Gateway controller* is responsible for intermediating management operations, such as performing polling on the *Agents* or monitoring and forwarding traps. In addition, according to where the *Virtual Network Node* is placed, *Gateways* may control other infrastructure's components directly or indirectly through the network. For example, switches installed with a *Virtual Network Node* can have their internal parameters changed by the *Gateway controller* locally, *e.g.,* changing the MTU and the routing protocol, or indirectly, by receiving remote requests through the network.

In summary, *Gateways* are important components to address the four MCC requirements. By intermediating communication with *Agents*, *Gateways* minimize network traffic and response time by being placed closer to the wireless infrastructure, partially tackling the *Performance* requirement. Also, when *Gateways* perform management operations, this alleviates the overall management task of the MCC environment that otherwise would have to be completely performed by *Management Applications*. Like in the Agent, *Security* is provided by the *AAA client*, while the *Configuration* requirement is met by the Service Interface. Finally, the *Fault* requirement is addressed by the *Gateway controller*, which monitors and processes the information sent by the *Agent System Extractor* to detect malfunctions as well as unavailabilities that degrades QoS and QoE.

3.3. Management application

An Administrator is not able to fully understand all the raw data gathered from the MCC environment without proper summarization, followed by the creation of comprehensive charts. Moreover, the Administrator cannot perform distribution of tasks inside an MCC environment without employing a distribution solution, *e.g.,* using management by delegation [20] or network function virtualization [21]. In addition, there are several management tasks that must be processed without the need for waiting for Administrators to make a proper decision. For example, when a critical trap is sent to inform about a link failure, the management system is supposed to perform an action to try to overcome this link failure, by preventing *Agents* from using the defective links. In this sense, data summarization and visualization , as well as distribution and processing of management tasks, are some of the responsibilities that a *Management Application* must perform in an MCC environment.

A *Management Application* is cloud software that can be placed in a *Virtual Cloud Node*, which has an OS and *Virtual Network Interface* that allow the communication with *Gateways* in an internal network. To perform all the aforementioned responsibilities, we propose a *Management Application* composed of seven modules, described below

- *Management Controller* is responsible for processing and distributing all the management tasks of the MCC environment. The controller can take advantage of the huge availability of *Cloud Resources* to employ different management algorithms for optimization purposes, *e.g.,* policy mechanisms, autonomous machine learning algorithms, and fuzzy logic. According to the employed algorithm, the MCC environment can be reorganized dynamically aiming for different objectives, for example, fairness or overall throughput improvement.
- *Service Interface* presents all the services available to be consumed for *Gateways* and other *Management Applications* in a standardized manner. In addition, all the distribution of management tasks among *Management Applications* is based on network services to provide decoupling and easy integration.
- *Graphic Unit* must continuously process all the data gathered by *Gateways* and *Agents* to create rich summaries, composing comprehensive charts or visual graphs. Through the visualizations created by this module, the *Management Application* allows Administrators to audit and monitor their MCC domains.
- *Report Unit* creates documents that summarize all the management performed inside the MCC domain. These documents can be periodically sent to Administrators for auditing or monitoring purposes. Also, inside a report, different management tasks are summarized in terms of traps received and what actions were performed by different entities.
- *Alert Unit* receives messages from the *Traps Unit* from *Gateways* and are responsible for warning Administrators through notifications, mainly when a critical event happened inside the MCC environment. In addition, the *Alert Unit* is responsible for summarizing the traps received and actions performed to be used later by the *Report* and *Visualization Units*.
- *AAA client* assures that all the gathered measurements are correctly binded to the End-users that collected it, for report generation purposes. Moreover, each role, *i.e.,* End-user or Administrator, which a user can assume when accessing the *Management Application* is granted by the *AAA client* module.
- *Web Application* is a web tool for Administrators and End-users to analyze reports and charts generated by the *Report* and *Graphic Units*, respectively. Also, Administrators can manage their MCC domains by auditing reports generated by the *Alert Unit*, and send management tasks to be performed by different *Gateways* or *Agents* manually if needed.

Management Applications play a key role in addressing three of the MCC management requirements. *Security*, addressed by *AAA client*, is the most important requirement for this component, since it allows grouping measurements binded to the End-user that collected it. In addition, it provides user access control in the *Web Application* module, granting protection for private data and different levels of permission for End-users and Administrators. The *Fault* requirement is met by analyzing charts and reports that include data collected from mobile devices. An Administrator can analyze this information to verify whether or not a given problem reported by an End-user is related to his/her mobile device. Similar to other components, *Configuration* is assured by the *Service Interface* module that guarantees interoperability among other components exposing configurable services to be used remotely.

4. CoLisEU

In this section, we describe the developed prototype, called CoLisEU, that was based on our architecture as a proof of concept. The scenario used for deployment and testing of the prototype is also detailed. Finally, metrics are defined for assessing the performance of CoLisEU.

4.1. Prototype and scenario

We developed the CoLisEU prototype considering the Agent, Gateway, and Management Application entities. The Agent was developed using Java and the software development kit provided by Google^{inc} for mobile devices with Android OS. A Gateway was developed as a network application over an Apache Web Server using PHP combined with a MySQL database. A virtual node was used to deploy the Management Application, which is an Asynchronous JavaScript and XML (AJAX)-based Web system. In such an application, reports and visualizations are processed using Web Service composition, *e.g.,* Google MAPs and Google Big Tables services.

The AAA server was developed as a database placed inside a virtual node on the cloud infrastructure. The Cloud Resource was implemented as a database to store QoS and QoE measurements. All the developed virtual nodes were placed over a private cloud infrastructure, based on a Xen server. The main developed modules for evaluating our prototype are QoS Meter, QoS Under Test, and the QoE Meter. The Agent QoS Meter and the Gateway QoS Under Test modules were both developed as part of the performance tools from Internet2 perfSONAR, OWPING, OWAMP, and BWCTL,[2] while the QoE Meter was developed using Java.

To evaluate the prototype, we defined two different test scenarios to understand how CoLisEU must be applied to real MCC environments. The first scenario regards an educational wireless network, *i.e.,* students from the University of Santa Cruz used their mobile devices to contribute QoS and QoE measurements on the evaluation of the university wireless network. On the other hand, the second scenario is composed of residential wireless networks, where End-users are at their own home. Both scenarios differ from each other, since the behavior and the expectation of the End-Users are different, affecting the subjective evaluations.

4.2. Evaluation methodology

A team composed of 68 volunteers was responsible for collecting QoS and QoE measurements for the two scenarios, educational and residential, using mobile devices such as

[2] http://www.internet2.edu/products-services/performance-analytics/performance-tools/ .

smartphones and tablets. Using the gathered measurements, we were able to evaluate the combination of QoS and QoE, making two contributions: (*i*) the assessment of the composed visualization and the management cost that CoLisEU inserts into the MCC environment; (*ii*) the establishment of the relationship between QoE and QoS to improve the management of an MCC environment. In the first contribution, three metrics were considered: the consumption of battery, network traffic, and response time for each measurement. These metrics were chosen because they directly impact on the End-user experience when using the MCC application.

As a baseline for comparison, we used a management operation with the minimum information required for a mobile device to communicate with a gateway, *i.e.,* username, password, and OS details. For each experiment performed, we conducted replications of 30 experiments to assess each measurement gathering. For each replication we presented the mean values with a 95% confidence level.

Our second contribution is deriving an equation for each scenario that uses QoS measurements to infer a subjective perception about the wireless infrastructure. Deriving these equations is important for a management system so it can predict the user experience through QoS metrics. This means that the management system can reconfigure the network with fewer QoE measurements. Such derivation was possible by applying a linear regression to the measurements collected by the volunteer teams in both evaluation scenarios. In the next section, we analyze the prototype, using the experimental results collected for the proposed scenarios.

5. Results & discussions

Since CoLisEU is concerned with the management of MCC considering both objective and subjective perspectives, we chose to focus on a qualitative analysis in Section 5.1 and on a quantitative analysis in Section 5.2.

5.1. Qualitative analysis

This analysis intends to explore the benefits acquired from using the combination of QoS and QoE in the management of MCC. To expose these benefits, we chose the main screen of our prototype, as shown in Fig. 4. This visualization composes a heat map showing the experiences that End-users had when using a federated network, in particular, the eduroam WiFi network. The colored amorphous shape in the map represents the estimated geographic coverage area of the analyzed network. In addition, the spreading gradient colors depict the average End-user experience within the selected network. CoLisEU End-users can interact with the visualization to analyze the nearest access point in the selected area, along with meaningful information about it. Examples of such information would be the IP address or geographical position of the access point, or even a throughput chart that allows End-users to realize the average throughput experienced by other users using the same access point. Furthermore, the side menu also enables the End-users to switch between networks and change the measurement time intervals being considered.

Based on the visualization provided by the prototype, we encourage some discussion on the advantages of managing MCC considering objective and subjective perspectives combined with rich visualizations.

- **Detecting End-users behavior** in MCC can potentially influence Administrators' decision-making about network configurations. QoE and QoS measurements can be combined to determine different End-user profiles, *e.g.,* End-users that are interested in games, video chats, or web navigation. Each profile may lead Administrators to configure their infrastructures differently, such as for example, creating policies that prioritize smaller round trip times, instead of improving link capacity.
- **Estimating the wireless network coverage** is essential for Administrators to reduce the cost of the wireless infrastructure, in terms of antennas and network devices. For the End-user, the main advantage could be the possibility of improving the connectivity achieved inside a covered area.
- **Summarizing the average satisfaction** within the network may be interesting for End-users to know what to expect when connecting in a given wireless network. In addition, this satisfaction could be used by an End-user to troubleshoot his/her mobile device.
- **Analyzing the infrastructure performance and the End-users' satisfaction** is possible through detailed reports, which contains data and charts. Synthesizing a huge amount of information in a comprehensive visualization allows the management system to be used either by expert Administrators or lay End-users.

5.2. Quantitative analysis

The developed prototype is a management system that enables the retrieval of QoE in terms of Mean Opinion Score (MOS), which is represented by the values 1 (very unsatisfied), 2 (unsatisfied), 3 (regular), 4 (satisfied), and 5 (very satisfied). QoS values are gathered in terms of RSSI, RTT, and throughput. However, gathering these measurements requires the consumption of resources from device, network, and cloud.

5.2.1. Resource consumption

For the quantitative analysis, we assessed the consumption of battery, network traffic, and the average response time for each measurement. The results are shown in Fig. 5. The battery consumption (*a*) was measured in milliamperes per hour [mA/h], the network traffic (*b*) in bytes, and the response time (*c*) in milliseconds [ms]. The measurements were organized and classified in four groups: Baseline only, Baseline and QoE, Baseline and QoS, and Baseline, QoS, and QoE simultaneously.

The summary presented in Fig. 5 allows us to evaluate the management cost for an MCC environment. From the network consumption point-of-view, gathering QoS measurements with our prototype is, in the worst case, 42% costlier than retrieving a QoE measurement. This difference is mainly due to the technique employed to measure QoS, *i.e.,* the network stress test. Considering the battery consumption, QoS costs 6% more than QoE, while considering the response time, QoS takes 252% more time to gather than QoE. These two results are related, since the QoS stress test takes much more processing time than simply informing a MOS.

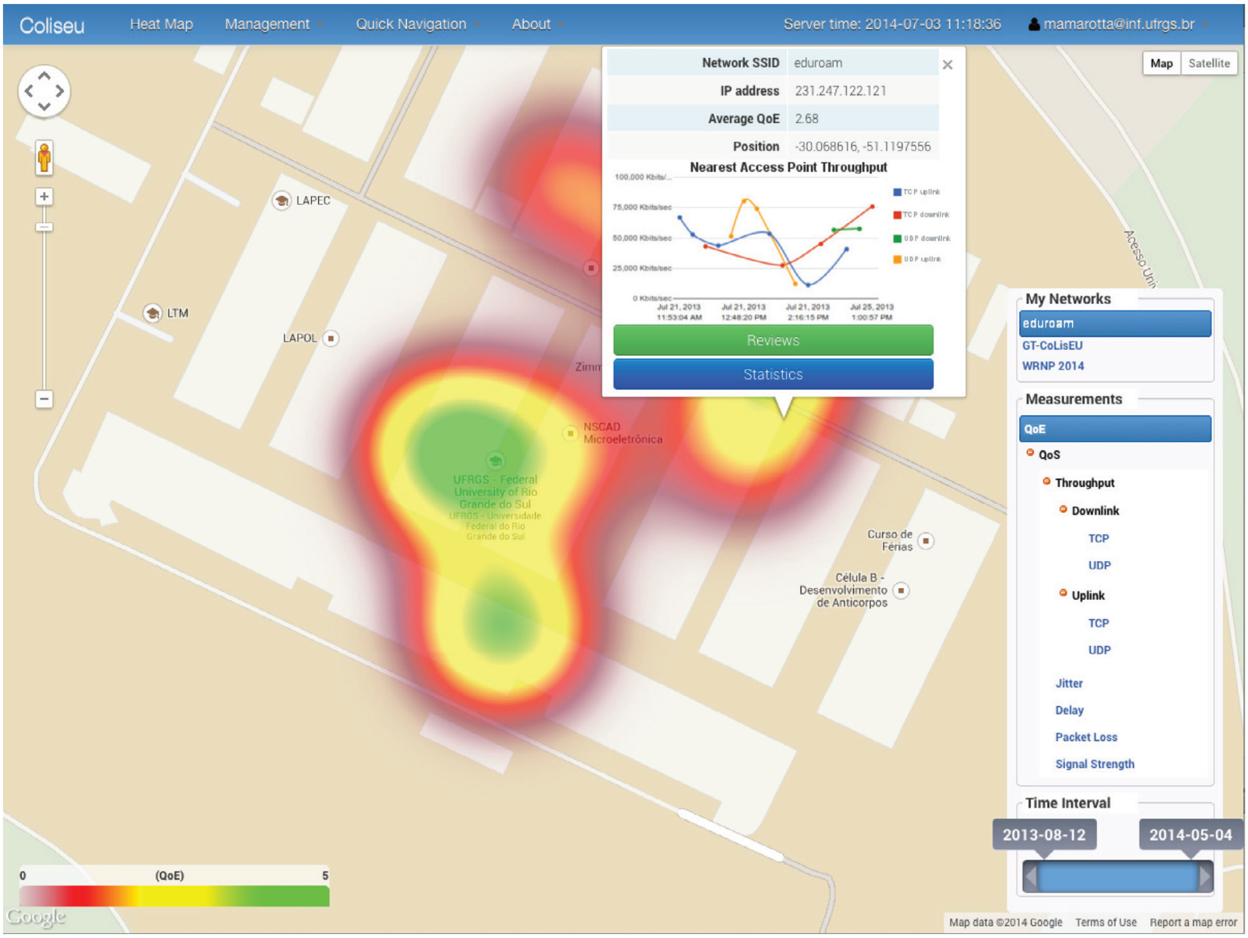
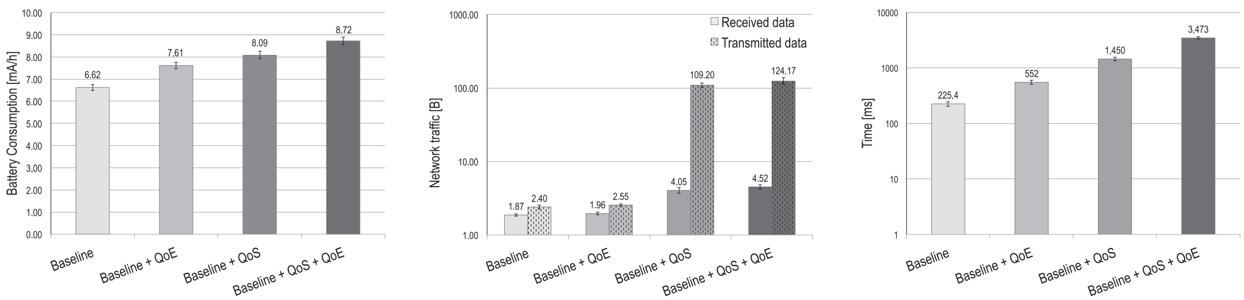


Fig. 4. Management application.



(a) Battery cost (b) Network Traffic (c) Response Time

Fig. 5. Prototype resource consumption evaluation.

Based on these results, it is possible to state that, from the End-user point-of-view, considering QoE measurements is important to avoid resource depletion with management operations. However, considering the Administrator point-of-view, the QoS measurements is the best way to detect specific infrastructure problems. By combining these two methods, the management system should explore the benefits of both, avoiding excessive costs. Therefore, the joint analysis of QoE and QoS improves the awareness of the MCC environment for both End-users and Administrators.

Our prototype showed that through the use of the designed model, it is possible to achieve the derived requirements to properly manage an MCC environment, *i.e.*, scalability, authentication, resource management, interoperability, and decoupling. In addition, charts and innovative visualizations were depicted according to the End-user geographic

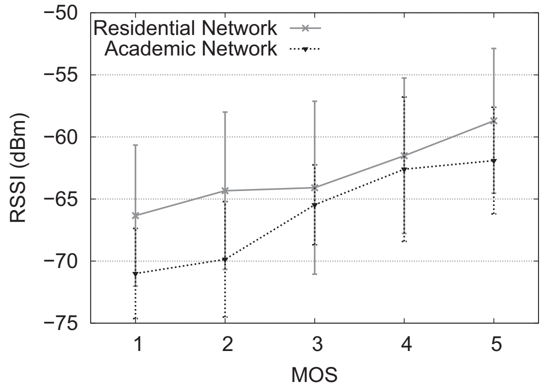


Fig. 6. RSSI versus QoE.

area, considering different authenticated Administrators. Moreover, the distribution of the management tasks, using Gateways, enabled trap detection for a larger number of End-users and network nodes from different infrastructures, without a single point of failure and network overload.

5.2.2. QoE versus QoS

The main idea of this subsection is to establish a relationship between QoE and QoS measurements, such as shown in Eq. 1. In order to establish this relationship, one thousand samples of QoE versus QoS were gathered in academic and residential scenarios. These samples were summarized in Fig. 7, according to three analyses: (*i*) the relationship between the *RSSI* versus the experiences of users (Fig. 6); (*ii*) the variation of *RTT* versus throughput (*TP*) for the experiences of academic users (Fig. 7a); and (*iii*) the same relationship of *RTT* versus throughput (*TP*) for the experiences of residential users (Fig. 7b); All of the analysis was carried out to investigate the importance of considering the satisfactions of users in different contexts for the management of MCC networks.

$$QoE = f(RSSI, TP, RTT) \tag{1}$$

As can be seen in Fig. 7, through QoS and QoE parameters we can establish the following relationships for a management system.

- The intensity of RSSI influences the user satisfaction, regardless of other QoS parameters for both networks.
- In an academic network where the users experience a low RTT, the main parameter that influences their satisfaction is the throughput because they expect a high data transfer rate. This influence becomes clear by the lightening of gradient colors that go along with the improvement of throughput as presented in Fig. 7a.
- In a residential network, which has an Internet access with limited bit rate according to a contracted SLA by users, the main QoS parameter that influences their satisfaction is the RTT. This influence becomes clear by the darkening of colors that go along with the RTT when it gradually gets larger, such as presented in Fig. 7b.

All of these relationships contribute to the definition of the function *f*(*RSSI*, *TP*, *RTT*). To make this definition, we used a multi-variable linear regression with our thousand samples of QoS and QoE, resulting in Eqs. 2 and 3. For each equation we found a significance level under 5%, proving their correctness for statistical purposes.

$$QoE = 5,435 + 0,06(RSSI) + 0,179(TP) - 0,00005(RTT) \tag{2}$$

In the academic network the *TP* is the most relevant QoS metric that influences the users' QoE. Through the Eq. 2, the *TP* presented a positive weight of 0,179 whereas the others presented less influence in the QoE summary. For a management system, this weight relationship among metrics is useful to coordinate each infrastructure of MCC to improve their link capacity, instead of, for example, coordinating flow priorities. In addition, this relationship also helps the University of Santa Cruz to direct their expenditure and, in this case, suggests to invest in wireless equipment in order to improve their link capacity with users.

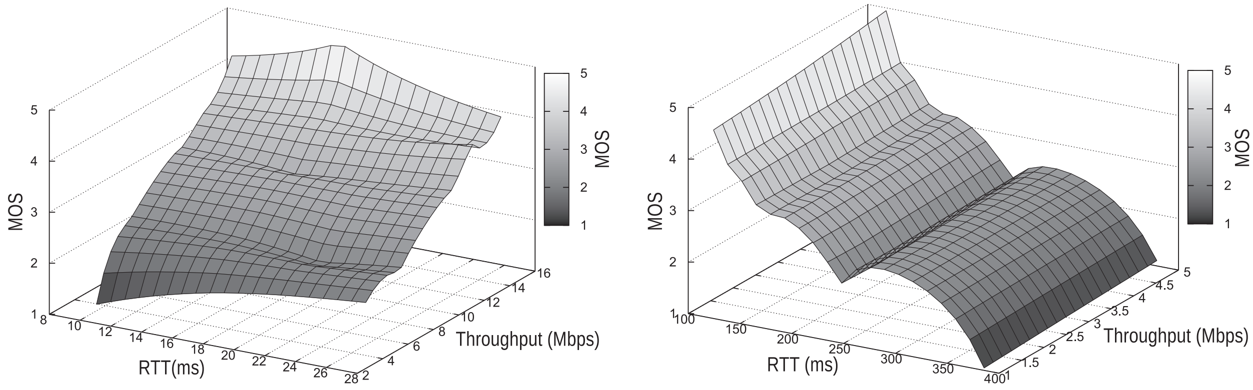
$$QoE = 2,762 + 0,002(RSSI) + 0,424(TP) - 0,018(RTT) \tag{3}$$

In residential networks, in turn, the Eq. 3 showed a higher weight for RTT than in an academic network. Despite the influence of RTT in residential networks, the Eq. 3 also showed that the *TP* also has a great influence in residential networks. The influence of *TP* was masked into the graphical charts, showing how important it is for a management system to calculate the linear regression. Using the Eq. 3, the management system must reconfigure the MCC environments to consider both, RTT and *TP*, as the main indicators for users' experiences. As a consequence, management systems will focus on guaranteeing the SLA for End-users, adding flows prioritizing minimizing RTT.

As can be seen, both equations are very different, which allows us to determine that users from different contexts have different QoS needs. This means that we cannot generalize this function for any context, making the gathering of QoE a neccessity for the management of the MCC environment according to different contexts. For the management system, it means that QoE values must be gathered and later clustered according to different contexts before the establishment of a relationship with QoS values.

Under another perspective, we tried to evaluate the End-user's point of view in academic and residential networks according to their most used type of applications, such as depicted in Fig. 8.

We considered five types of End-user applications; browsers, e-mail, games, chat, and social networks. End-users interested in playing games, using chats, or social networks, are susceptible to become dissatisfied and give a bad review for an academic network. Such a dissatisfaction might be related to the use of firewalls or network configurations that prevent the full usage of these type of applications in an academic institution. End-users in residential networks have their satisfaction less influenced by the type of application in use. Therefore, there will be cases where the same applications will influence the End-user satisfaction differently within the MCC environment. This means that a management system cannot consider just the type of applications in use when managing an entire MCC domain. The management of MCC must be performed through the gathering of QoE, QoS, and information concerning



(a) Academic network

(b) Residential network

Fig. 7. QoS versus QoE values in academic and residential scenarios.

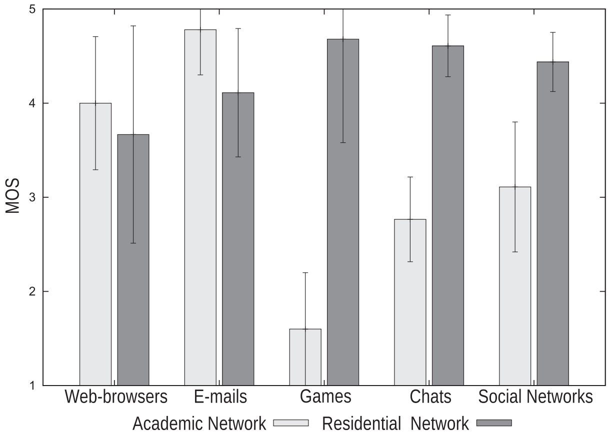


Fig. 8. QoE according to different users interests.

End-users' applications to be later clustered to find the real source of End-users' dissatisfaction managing each part of the MCC environment.

6. Conclusion & future work

In this paper we outlined MCC as the process of extending the cloud for resource augmentation of constrained mobile devices. Managing the infrastructures involved in the process of augmentation is fundamental to the MCC environment to properly operate. Administrators and End-users can collaborate with the MCC management by sharing their experiences through the use of a survey with detailed reviews. In addition, End-users can perform QoS measurements, enabling problem detection in the MCC environment. A model, architecture, and prototype were described to enable the collaboration between Administrators and End-users within the management of MCC, creating rich visualizations to be analyzed with regards to MCC performance.

The model and architecture were designed to meet MCC requirements integrating mobile devices, the MCC infrastructures, and the cloud. Our prototype, called CoL-isEU, validates the architecture and is divided into three components. Afterwards, we analyzed the prototype qualitatively and quantitatively, discussing the benefits of using objective and subjective perspectives for the management of the MCC environment. Combining both perspectives improves the awareness of Administrators and End-users about the MCC environment.

Our research is intended to be an initial step toward the introduction of a management system for MCC environments. As MCC develops, there will likely be other interesting research opportunities in this area. For example, semantic Web technologies can be used to improve the extraction of management information from QoE reviews. Finally, a management system could employ machine learning techniques to adapt these requirements to each different infrastructure of an MCC environment.

Acknowledgments

The authors thank to the Rede Nacional de Ensino e Pesquisa (RNP), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Coordenação de Aperfeiçoamento de Nível Superior (CAPES) for the financial support.

References

[1] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, R. Buyya, Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges, IEEE Commun. Surveys Tutorials 16 (1) (2014) 337–368.
[2] M.R. Rahimi, J. Ren, C.H. Liu, A.V. Vasilakios, N. Venkatasubramanian, Mobile cloud computing: a survey, state of art and future directions, Mobile Netw. Appl. 19 (2) (2014) 133–143.
[3] D. Huang, Mobile cloud computing, Multimedia Commun. Techn. Committee IEEE Communications Soc. E-Letter 6 (10) (2011) 27–30.
[4] F. Sardis, G. Mapp, J. Loo, M. Aiash, A. Vinel, On the investigation of cloud-based mobile media environments with service-Populating and QoS-aware mechanisms, IEEE Trans. Multimedia 15 (4) (2013) 769–777.
[5] P. Rengaraju, C.-H. Lung, F. Yu, A. Srinivasan, On qoe monitoring and e2e service assurance in 4g wireless networks, Wirel. Commun. IEEE 19 (4) (2012) 89–96.
[6] T. Hobfeld, R. Schatz, M. Varela, C. Timmerer, Challenges of qoe management for cloud applications, IEEE Commun. Magaz. 50 (4) (2012) 28–36.
[7] P. Casas, R. Schatz, Quality of experience in cloud services: survey and measurements, Comput. Netw. 68 (5) (2014) 149–165.

[8] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J.-H. Hong, A.K. Dey, Factors influencing quality of experience of common;y used mobile applications, IEEE Commun. Magaz. 50 (4) (2012) 48–56.
[9] J. Wan, Z. Liu, K. Zhou, R. Lu, Mobile cloud computing: application scenarios and service models, in: Proceedings of the 9th International Conference Wireless Communications and Mobile Computing, 2013, pp. 644–648.
[10] D. Huang, T. Xing, H. Wu, Mobile cloud computing service models: a user-centric approach, IEEE Network 27 (5) (2013) 6–11.
[11] Y. Cai, F. Yu, S. Bu, Cloud computing meets mobile wireless communications in next generation cellular networks, IEEE Network 28 (2014) 54–59.
[12] M. Santos, J. Villalon, L. Orozco-Barbosa, A novel QoE-aware multicast mechanism for video communications over IEEE 802.11 WLANs, IEEE J. Select. Areas Commun. 30 (7) (2012) 1205–1214.
[13] R. Boutaba, A. Polyrakis, Projecting fcaps to active networks, in: Proceedings of international Conference on Enterprise Networking, Applications and Services, 2001, pp. 97–104.
[14] A. Khan, M. Othman, S. Madani, S. Khan, A survey of mobile cloud computing application models, IEEE Commun. Surveys Tutorials 16 (1) (2014) 393–413.
[15] A.N. Toosi, R.N. Calheiros, R. Buyya, Interconnected cloud computing environments: challenges, taxonomy, and survey, ACM Comput. Surveys 47 (1) (2014) 7:1–7:47.
[16] B. Parak, Z. Ustr, Challenges in achieving iaaS cloud interoperability across multiple cloud management frameworks, in: Proceedings of the 7th International Conference on Utility and Cloud Computing IEEE/ACM, IEEE, 2014, pp. 404–411.
[17] H.T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches, Wirel. Commun. Mobile Comput. 13 (18) (2013) 1587–1611.
[18] N. Fernando, S.W. Loke, W. Rahayu, Mobile cloud computing: a survey, Elsevier Future Generation Comput. Syst. 29 (1) (2013) 84–106.
[19] H. Wu, D. Huang, S. Bouzefrane, Making offloading decisions resistant to network unavailability for mobile cloud collaboration, in: Proceedings of IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2013, pp. 168–177.
[20] M.A. Marotta, F.J. Carbone, J.J.C. de Santanna, L.M.R. Tarouco, Through the internet of things – a management by delegation smart object aware system (MbDSAS), in: Proceedings of the 37th Annual Computer Software and Applications Conference, 2013 IEEE Kyoto, Japan, 2013, pp. 732–741.
[21] M. Peng, Y. Li, Z. Zhao, C. Wang, System architecture and key technologies for 5g heterogeneous cloud radio access networks, IEEE Network 29 (2) (2015) 6–14.

Marcelo A. Marotta is a Ph.D. student in Computer Science at the Institute of Informatics (INF) of the Federal University of Rio Grande do Sul (UFRGS), Brazil. He received his M.Sc. title in (2013) from INF of UFRGS, Brazil. In addition, he holds a B.Sc.in Computer Science from the Federal University of Itajubá (UNIFEI) (2010), Brazil. His research involves Wireless Networks, Next Generation Networks, Internet of Things, Software Defined Radio, Cognitive Radio Networks, and Heterogeneous Cloud-Radio Access Networks.

Leonardo R. Faganello is an M.Sc. student at the Federal University of Rio Grande do Sul (UFRGS), in Brazil. He achieved his B.Sc. degree in Computer Engineering at UFRGS in 2012. His current research interests include wireless networks, cognitive radio networks and spectrum decision.

Matias Artur Klafke Schimuneck is an M.Sc. student at the Federal University of Rio Grande do Sul (UFRGS), in Brazil. He achieved his B.Sc. degree in Computer Science at University of Santa Cruz do Sul in 2014. His current research interests include wireless systems and users quality of experience.

Lisandro Z. Granville is an associate professor at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil. He received his M.Sc. and Ph.D. degrees, both in computer science, from UFRGS in 1998 and 2001, respectively. He is member of the Brazilian Internet Committee (CGI.br). He has served as a TPC member for many important events in the area of computer networks (e.g.,IM, NOMS, and CNSM) and was TPC co-chair of DSOM 2007 and NOMS 2010.

Juergen Rochol is an associate professor at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil. He received his M.Sc. degree in Physics, and his Ph.D. degree in Computer Science, both from UFRGS in 1972 and 2001, respectively. His research interests include wireless networks, next generation networks, optical networks and traffic control on broadband computer networks.

Cristiano B. Both is a Postdoctoral in Computer Science at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil. He received a Ph.D. degree at UFRGS in 2011 and his M.Sc. degree in Computer Science from the Pontifical Catholic University of Rio Grande do Sul, Brazil, in 2003. He is currently an assistant professor at the Federal University of Health Sciences of Porto Alegre, Brazil. His research interests include wireless networks, next generation networks, and software-defined networking.