Impact of Fog and Cloud Computing on an IoT Service Running over an Optical/Wireless Network Testbed

Aloizio P. Silva†, Bernardo A. Abreu†, Erik B. Silva†, Marcos Carvalho†, Matheus Nunes†, Marcelo Marotta§, Ali Hammad*, Carlos F. M. Silva†, João F. N. Pinheiro†, Cristiano B. Both§, Johann M. Marquez-Barja*, and Luiz A. DaSilva†

†Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil
{aloizio,bernaardoabreu,erik,mhnunnnes,marcoscarvalho}@dcc.ufmg.br, erik@decsi.ufop.br
§Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
{mamarotta,ceboth}@inf.ufrgs.br
*CONNECT Centre for Future Networks and Communications, Trinity College Dublin, Ireland
{Johann.Marquez-Barja,dsilval}@tcd.ie
†Federal University of Ceará (UFC), Fortaleza, Brazil
{cfms,joao}@gtel.ufc.br
‡University of Bristol (UNIVBRIS), Bristol, United Kingdom
ali.hammad@bristol.ac.uk

Abstract—With the advance of the Internet of Things (IoT), the interaction between humans and smart objects is already a reality. New applications that are expected to operate in dynamic environments must support different modes of human/machine interaction (e.g., voice and sign language), exhibit same or better performance in heterogeneous wireless and optical networks, and be able to react in real time. In particular, dispersed computing has arisen as an approach to deal with latency issues in this context. In the work described herein, we design a smart lighting IoT system that allows control of light bulbs (turn on/off, color and brightness change) through voice and sign language. This work addresses the idea of dispersed computing, which is implemented through fog computing, and combines it with virtualized resources to mitigate latency in the convergence point between wireless and optical networks. The proof-of-concept implementation of our design demonstrates the viability of the approach.

I. INTRODUCTION

Cloud Computing (CC) is a paradigm that integrates cloud and edge devices, such as computers and smart phones as well as sensors and actuators of the Internet of Things (IoT). CC allows the augmentation of constrained resources in edge devices, such as processing, storage, and battery autonomy, by using cloud services [1]. The augmented resources available at the cloud enable edge devices to execute more sophisticated versions of key applications and services, such as mobile learning, e-commerce, and sound/image recognition [2]. Tasks in CC are partially performed in edge devices and partially computed in the cloud. Thus, the network that interconnects edge devices and the cloud impacts the proper execution of CC applications and services. This network is usually composed of wired segments in the core, using a fiber optical medium, and wireless links at the edge to communicate with mobile devices. We refer to this combination of wireline and wireless network resources as converged networks. In this work, we discuss the allocation of computing and networking resources in a converged network, illustrating the relevant issues through the testbed implementation of a smart lighting IoT application.

To meet the stringent latency requirements of real-time services in converged networks, the processing power must be distributed throughout the infrastructure, in what is widely known as fog computing. In fog computing the processing resources can be centralized in the cloud or distributed within the network through the usage of Micro Data Centers (MDC) that are closer to the network edge. The MDC proximity to the edge decreases the usage of core infrastructures, such as optical links, and the network latency by providing a closer pool of processing resources able to process edge devices’ and smart objects’ applications workload. Decreasing the network latency is crucial for real-time applications [3] whose performance is measured in terms of response time, e.g., the time from issuing a voice/sign command until the real time system performs an action in the users’ environment, such as turning on the lights in a room.

The performance trade-offs of using fog computing in support of latency-sensitive network applications remains insufficiently explored in the literature. In this paper, we assess the employment of fog computing in converged networks to meet quality of service (QoS) requirements of real-time systems, highlighting its pros and cons. The application we selected is a real-time service for people with special needs, who can use speech and/or sign language to issue commands to a smart lighting system. We use CoLisEU [4], a management tool for network infrastructures, in conjunction with Internet 2 performance tools such as the Bandwidth Test Controller (BWCTL) and One-Way Ping (OWAMP), to measure the latency of the converged network. The response time of the
application is measured from the time a voice/sign language command is captured by a sensor until the smart light controller performs an action. Finally, we compare the different positioning of the processing resources inside the network, by changing the fog tiers being used and assessing its impact on real-time applications in converged networks.

II. RELATED WORK

There are several factors that affect IoT scenarios, such as devices’ features, geographical location, and network architecture. Yannuzzi et al. [5] conclude that the current datacenters’ locations will not be able to fulfill the requirements of foreseen IoT applications. Considering mobility, reliable control, and scalability, the authors analyze fog computing as a natural platform for IoT, also addressing the increasing importance of the interplay between fog and cloud in coming years. Bibani et al. [6] propose a hybrid fog/cloud architecture in order to provide a feasible IoT solution for low-latency applications such as fire fighting. Authors demonstrate their approach, providing evidence that fog and cloud computing can alternate their computing role within the hybrid architecture to meet different application requirements; in particular, to reduce latency, the architecture can allocate the computing processes in the fog, closer to IoT devices. Confirming these findings, Shi et al. [7] present a solution called Fog-Radio Access Network (F-RAN) which is able to bring efficient computing closer to IoT devices. This experimentation-based paper.

III. ARCHITECTURE

For our experimental assessment of the locus of computing in a converged wireless/optical network, we have built a smart lighting system in which a person can control the lights through either voice or sign language commands. In particular, the lights can be controlled on a coarse three levels: turn on/off, increase/decrease brightness, and change color.

This smart lighting system is implemented in the context of the FUTEBOL project [9], which establishes testbeds and a control framework in Europe and Brazil to conduct experimental research on wireless and optical networks. The experimental platform includes federated research facilities on both sides of the Atlantic, allowing experimental research collaborations at the convergence point of wireless and optical networks.

Figure 1 shows the smart light system architecture and its main components. This system spans a wireless and an optical network and is divided into two segments. The first operates in the WiNet Laboratory (DCC-UFMG, in Brazil). In this segment, the command (voice or sign language) is captured. The second segment operates across the optical network at the University of Bristol (UNIVBRI), in the UK. This is where the captured command is processed. In this case, the processing takes place in the cloud. However, the second segment can still be operated either in a local server or in a gateway. Thus the processing takes place either locally or in a fog.

The architecture is composed by four building blocks (see Figure 1) as described below.

1) IoT Application/Demo: the IoT application is implemented using two Raspberry Pi 3 Model B (actuator and sensor) running Raspbian Jessie (an operating system developed for Raspberry Pi based on Linux Debian Jessie) and a set of three smart light bulbs and their controller. The microphone and the Leap Motion device are used to capture the voice and sign language commands, respectively. The light bulbs can be controlled through voice or sign language (in the latter, commands are issued using one’s hands). For the voice recognition component of the system, we used CMU Sphinx [10] version pocketsphinx 0.1.3, as it provides an open source Application Programming Interface (API).

2) FUTEBOL Facility [9]: this block contains three Dell Alienware Alpha Mini Gaming PCs, running Ubuntu 16.04.1 x64 bits, which connect with two Raspberry Pi 3 Model B through a MikroTik Wireless Access Point (AP). This AP is connected to a Pica 8 Switch running OpenVSitch version 2.3. The Dell MiniPCs are running a Virtual Machine created using the Cloud Based Testbed Manager (CBTM), a software designed for creating and managing virtual machines on remote hosts.

3) FIBRE Island: it is composed by three PRONTO Switches, a Pica 8 Switch and a VM Server. The FIBRE [11] Island is connected with the FUTEBOL Facility which is federated by Fed4FIRE [12] allowing the connection with the optical network in UNIVBRI.

4) Cloud/UNIVBRI: this block includes all the optical infrastructure where VMs running Docker [13] at UNIVBRIS process the voice or sign language commands.

Note that our system can be connected to other IoT devices in future smart buildings or smart homes. At present, we run the demonstration system in a wireless network research laboratory to explore IoT applications in a controlled environment.

IV. METHODOLOGY

The main goal of our experimental system is to evaluate, by means of different fog computing tiers, the performance of the IoT application in terms of latency in system response to either voice or sign language commands being issued. Specifically, the voice and sign command processing can be performed in different fog computing tiers: 1) tier 0 is the MiniPC/Gateway (VMLocal); 2) tier 1 is the FUTEBOL facility Server (VMServer); and 3) tier 2 is the remote cloud at
UNIVBRIS (Cloud). Using the ColisEU [14] tool, the end-to-end latency is measured in real time in both segments, wireless and optical. The measurement starts from the moment that a command is captured and processed, until the moment when the light bulb reacts (i.e., changes its color or its brightness intensity). We use Docker version 1.5.1 to deploy the main routines in the optical segment; Ubuntu version 14.04 is used as the operating system for the Virtual Machine (VM). The VMs created in VMLocal and VMServer have Ubuntu Server 16.04.1, 4GB of RAM, 4 virtual cores and 20GB disk space. The VMs created in the Cloud have Ubuntu Server 16.04, 2GB of RAM, 2 virtual cores and 40GB disk space.

A. IoT Scenario Overview

The ability to control the lights in a home or building without the need of a physical light switch is particularly important in the context of smart homes and accessibility to disabled persons. In this section we present a self-contained and connected smart lighting prototype that operates over a wireless/optical converged network. Figure 2 shows the testbed environment setup. Specifically, this picture captures a sign language command being used to “turn on” the lights. The communication among the devices is wireless (the two Raspberry PI 3 Model B, the lights controller and the lamps themselves, the LeapMotion and the MiniPCs). Two videos that show the demonstration working in real time can be found in YouTube 1. The code we used is available in Bitbucket2.

---

1 Sign language: https://www.youtube.com/watch?v=_RcrSZW3yrQ
Voice: https://www.youtube.com/watch?v=snh9yAB7w1Y.
2 Sign language: https://bitbucket.org/futebolUFMG/leap-motion
Voice: https://bitbucket.org/futebolUFMG/voice_control
V. TESTS AND ANALYSIS

The main goal of the prototype system is to show that both voice and sign language commands can be continuously and reliably carried via wireless/optical network links to be processed in different places in the network. We measure network latency and identify its impact on the response time of voice/sign language recognition systems in a CC converged network. Also, the idea of dispersed computing, represented in this paper by fog/cloud computing, allows us to bring the computation closer to the user, resulting in a reduced response time. This section summarizes the performance results, characterized in terms of latency, collected in our experiments.

A. Network characteristics

Figure 5 shows the latency observed when the processing is performed in the cloud (first 100s), and in the VM Server (last 100s).
in Figure 5 are related to the fluctuations in propagation delay, routing and switching, and queuing and buffering. In spite of being in a controlled experimental environment, the wireless network still suffers some interference. According to [3], a latency higher than 150 ms is already annoying for typical real-time users. In this case, the usage of the VM Server becomes a must to avoid the network latency to impact negatively in the user experience.

B. Fog computing for sound recognition systems

To fully exploit the impact caused by adding fog computing in a CC environment, we first analyze the time consumed to process voice commands in different processing tiers (or MDC), i.e., the Cloud, VM Server, and a Local VM. These voice commands change in terms of processing complexity by adding new operations to a single recording, i.e., (a) change the light colors, (b) turn on the lights, (c) turn on the lights with a specific color, (d) turn on a specific light bulb with a determined color, and (e) turn on a specific light bulb with a determined color and brightness. Afterwards, we evaluate the time consumed for processing each of these commands adding the impact caused by the data traversing the network with the response time for different processing tiers. It is important to notice that the cloud processing resources were restrained to emulate a situation where the processing resources are under heavy usage requiring to move the processing to the edge where the other tiers are placed.

![Figure 6. Processing time for different voice commands.](image)

In Figure 6, the processing time obtained by each processing tier is measured and depicted. In the y-axis the time consumed to process each voice command for different processing tier is represented in milliseconds (ms), whereas in the x-axis each voice command is presented from the simplest to the most complex processing case. For the simplest voice command the cloud under heavy usage presented the worst processing time, whereas the VM Server and Local VM presented smaller processing times. This can be explained by the fact that the Cloud has been overloaded. For the most complex command, Cloud, VMServer, and Local do not present significantly different latency, due to their error bars being overlapping. It means that, even during heavy load, as the cloud has more robust processing power, it is able to process the more complex commands almost with the same time as the other tiers.

![Figure 7. Response time for different voice commands.](image)

In Figure 7, the response time measured for each processing tier is depicted. In the y-axis the response time for each voice command per processing tier is represented in ms, whereas in the x-axis each voice command is presented from the simplest to the most complex processing case. For different voice commands the response time is always longer when the voice recognition processing is performed in the cloud, as compared to either VMLocal or VM Server. For increasingly complex commands, the response time grows significantly, mainly for the Cloud, in the worst case reaching 1190 ms to perform the “light bulb two on color blue” command. For the VM Server and the Local processing tiers, the wireless environment itself caused fluctuations in the performance of all commands, with overlapping error bars for response time.

![Figure 8. Response time for different sign language commands.](image)

In Figure 8, the response time measured for each processing tier is depicted. In the y-axis the response time for each sign language command per processing tier is represented in ms, whereas in the x-axis each sign language command is presented from the simplest to the most complex processing case. As expected the response time when sign language processing takes place in the Cloud is greater than the response time in the VMLocal and VM Server. Note that the response times for the VMLocal and VM Server are approximately the same when the standard deviation is considered. This can be explained by the fact that both MDC are located in the
proximity of the FUTEBOL facility and they are subject to the same environmental factors e.g., congestion and propagation delay.

Note that the results shown in Figures 7 and 8 illustrate the response time for voice and sign language commands, respectively. It is interesting to highlight that the voice commands’ response time is greater than the sign language’s response time. This happens because the audio file has greater size than the sign language file, since in the last case the hand movement is codified in a string vector to be sent through the network. As a result the file size is small, contributing for a small response time.

VI. CONCLUSIONS

Ubiquitous human interaction with smart objects in dynamic environments is one of the advances promised by IoT systems. In this human-smart object interaction, latency is a critical QoS parameter because it directly affects the human perception that an action has taken place. This problem becomes even more serious in critical real-time applications.

By exploiting the concepts of converged networks (i.e., the interplay between wireless and optical networks) and dispersed computing, the information may be processed in different network tiers (i.e., cloud and fog) and such processing can be adjusted according to a set of service and application requirements.

In this paper we assessed, in terms of response time and processing time, the use of fog computing to control a smart lighting IoT system that enables the control of the lights of a room, through speech and sign language, especially helpful for disabled people. From the processing time point of view, for processing complex voice commands, we cannot see a big difference in latency between the cloud, VMLocal or VMServer (see Figure 6). As for simple commands, the processing time is smaller in the tiers closer to the mobile device (local machine or VMServer). Although the cloud could have more processing power, it results in longer latency when there is competition for network resources. And last, from the response time point of view, a combination of network latency and processing is assessed. Using voice processing, we have close response times for simple or complex voice commands, when comparing the response times on the same tiers (see Figure 7). When comparing voice to sign language command processing (see Figure8), we have a big difference on the response times. It is caused specially due to the data processed. Audio files are big compared to the vectorized data of the hand gestures. Also, the audio processing uses voice recognition, which is a heavy duty process, whereas the gesture processing, using vectorized data processing, represents a lighter type of processing.

Some of the future works may focus on:

- The performance of different voice recognition software in our IoT application;
- The impact of using different hand movements to activate the lights so that usability levels are measured for disabled people;
- The impact of VM migration in different network tiers;
- The integration of IoT Demo and CoLiSEU to measure latency and delay in real-time.
- The study of sensitivity analysis where depending on the resource availability at different tiers and propagation delays the smart lighting system will choose the tier that provides the shortest response time.

ACKNOWLEDGMENTS

This work has received funding from the European Union’s Horizon 2020 for research, technological development, and demonstration under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology and Innovation (MCTI) through RNP and CTIC.

REFERENCES