FEDERAL UNIVERSITY OF PARÁ
INSTITUTE OF TECNOLOGY
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

HUGO LEONARDO MELO DOS SANTOS

# A MULTI-TIER FOG ARCHITECTURE FOR VIDEO ON DEMAND STREAMING

**DM:** 11/2018

**UFPA / ITEC / PPGEE**

**Guamá University Campus**

**Belém-Pará-Brazil**

**2018**

FEDERAL UNIVERSITY OF PARÁ
INSTITUTE OF TECNOLOGY
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

HUGO LEONARDO MELO DOS SANTOS

# A MULTI-TIER FOG ARCHITECTURE FOR VIDEO ON DEMAND STREAMING

Dissertation submitted to the judging panel at the Federal University of Pará as part of the requirements for obtaining a Master's Degree in Electrical Engineering in the area of Applied Computing.

**UFPA / ITEC / PPGEE**
**Guamá University Campus**
**Belém-Pará-Brazil**

**2018**

Dados Internacionais de Catalogação na Publicação (CIP)
Sistemas de Biblioteca da UFPA

---

Santos, Hugo Leonardo Melo dos, 1990-
A Multi-tier Fog Architecture for Video On Demand Streaming / Hugo Leonardo Melo dos Santos. - Belém-Pará-Brazil, 2018.

Orientador: Eduardo Coelho Cerqueira

Dissertação (Mestrado) - Universidade Federal do Pará. Instituto de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2018.

1. Computação em nuvem 2. Sistemas multimídia 3. Dispositivos de redes sem fio I. Título.

CDD 22.ed. 004.6782

---

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

# "A MULTI-TIER FOG ARCHITECTURE FOR VIDEO ON DEMAND STREAMING"

AUTOR: **HUGO LEONARDO MELO DOS SANTOS**

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 05/03/2018

**BANCA EXAMINADORA:**

Eduardo Cerqueira
______________________________
**Prof. Dr. Eduardo Coelho Cerqueira**
(Orientador – PPGEE/UFPA)

Diego Lisboa Cardoso
______________________________
**Prof. Dr. Diego Lisboa Cardoso**
(Avaliador Interno – PPGEE/UFPA)

Denis Lima do Rosário
______________________________
**Prof. Dr. Dênis Lima do Rosário**
(Avaliador Externo ao Programa – PPGCC/UFPA)

**VISTO**:

______________________________
**Prof.ª Dr.ª Maria Emília de Lima Tostes**
(Coordenadora do PPGEE/ITEC/UFPA)

*I devote this work first to my family and friends, especially my parents, Emanuel and Luiza, who throughout my life loved me and supported me for the outcome of this stage of my life.*

*Dedico este trabalho primeiramente à minha família e amigos, especialmente meus pais, Emanuel e Luiza, que durante toda a minha vida me amaram e me prestaram o suporte para o desfecho desta etapa de minha vida.*

# Acknowledgement

I dedicate my sincere thanks to:

–To the Professor Dr. Eduardo Cerqueira, for the guidance, encouragement and for the confidence that he placed in me.

–To Professor Denis Rosário, for the dedication and effort offered to make this work possible.

–To the professors Doctor Cristiano Both and Jéferson Noble, for the hours in meeting in the Hangout to discuss about this work.

–To the students of UFRGS Matias Schimuneck and João Camargo, for the help in the experiments in the testbed of the Fiber.

–The team of the laboratory Wellington, Pedro, Fábio, Felipe, Danileno and Arnaldo, especially by the colleagues Iago and Joahannes, by the aid in diverse moments.

–To CAPES for offer the scholarship

*"happiness only real when shared"*
***Alexander Supertramp***

# Abstract

Abstract of Dissertation presented to UFPA as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

## A Multi-tier Fog Architecture for Video on Demand Streaming

Advisor: Eduardo Coelho Cerqueira
Cloud computing: Multimedia Systems; Network Wireless Devices.

Users are changing their traditional communication paradigm based on voice calls or text messages to real-time or on demand video services consumed on mobile devices. In this sense, the transmission of video content considering an adequate Quality of Experience (QoE) in mobile wireless networking infrastructures is a critical issue in both academic and industrial communities. Furthermore, video on demand have a growing consumption over Internet requiring higher bandwidth and lower latency. In this context, a fog computing paradigm can enhance the user experience in wireless networks. Fog computing for video on demand streaming can improve QoE by both video caching and adaptation schemes. However, it is important to evaluate the performance of downloading the videos with different codec configuration and cached closer to the user to measure the gain from the user perspective. We designed a multi-tier fog computing architecture with three levels located in the cloud, nearer the edge and in mobile devices. We evaluated the performance of downloading the video from multiple tier located in distinct geographical with a multimedia application. We assessed in an experimental environment with idle and congested network of streamed videos coded into H.264 and H.265 with three bitrates in a scenario deployed in the FIBRE testbed. We collected QoE metrics, playback start time, freeze times, QoS metric, round-time trip, and energy consumption to analyze the gains for each video configuration. These results showed an important understanding about cache, codec and bitrate schemes in multimedia networking scenarios.

# Resumo

Resumo da Dissertação apresentada à UFPA como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

## A Multi-tier Fog Architecture for Video On Demand Streaming

Orientador: Eduardo Coelho Cerqueira
Computação em Nuvem: Sistemas Multimídia; Dispositivos de Redes sem Fio

Os usuários estão mudando seus hábitos de comunicação tradicional com base em chamadas de voz ou mensagens de texto para serviços de vídeo em tempo real ou sob demanda consumidos em dispositivos móveis. Nesse sentido, a transmissão de conteúdo de vídeo considerando uma Qualidade de Experiência (QoE) adequada nas infraestruturas de redes sem fio móvel é um problema crítico nas comunidades acadêmicas e industriais. Além disso, o vídeo sob demanda tem um consumo crescente na Internet, exigindo maior largura de banda e menor latência. Neste contexto, um paradigma de computação em névoa pode ser aplicado para melhorar a experiência do usuário em redes sem fio. A computação em névoa pode melhorar a QoE para transmissão de vídeo sob demanda, tanto fazendo cache de vídeo quanto fazendo adaptação de conteúdo na borda da rede. No entanto, é importante avaliar o desempenho do download de vídeos com diferentes configurações de codec e em cache mais próximo do usuário para medir o ganho sob perspectiva do usuário. Projetamos uma arquitetura de computação em névoa de várias camadas com 3 níveis localizados na nuvem, um perto da borda e outra nos dispositivos móveis. Avaliamos o desempenho do *download* do vídeo de vários níveis localizados em aplicações geográficas distintas para aplicações multimídia. Nós avaliamos em um ambiente experimental com rede livre e congestionada de vídeos transmitidos codificados em H.264 e H.265 com 3 taxas de bits em um cenário implantado no testbed FIBRE. Nós coletamos métricas de QoE, tempo de inicial para reprodução, tempo de congelamento, métricas de QoS, tempo de viagem ida e volta do pacote e consumo de energia para analisar os ganhos para cada configuração de vídeo. Esses resultados mostraram uma compreensão importante do uso de cache , codec e taxa de bits em cenários de redes multimídia.

# Contents

# List of abbreviations

| | |
|---|---|
| VoD | Video on Demand |
| QoE | Quality of Experience |
| D2D | Device-to-Device |
| FN | Fog Node |
| QoS | Quality of Service |
| codec | Coder-Decoder |
| MPEG | Motion Experts Group |
| QoS | Video Coding Experts Group |
| IETF | Internet Engineering Task Force |
| NQoS | Network QoS |
| PQoS | Perceived QoS |
| HTML | HyperText Markup Language |
| HD | High Definition |
| HABS | HTTP Based Adaptive Streaming |
| DASH | Dynamic Adaptive Streaming over HTTP |
| BS | Base Station |
| RAN | Radio Access Network |
| ICN | Information Centric Networking |
| AP | Access Point |
| CDN | Content Delivery Network |
| EPC | Evolved Packet Core |
| RTT | Round-Time Trip |

# List of Figures

# List of Tables

# CHAPTER 1

# Introduction

This chapter introduces concepts about fog computing, highlighting the current main challenges for streaming of Video on Demand (VoD), outlines essential contributions and presents the content of the next chapters of this future dissertation work.

## 1.1 Overview

The smartphones managed to play videos almost in any place and end users started to play even more videos when they could download not only from Wi-Fi but also from mobile networks. Besides, multimedia distribution gained new markets beyond VoD with live streaming on popular streaming web sites, such as YouTube live[1] and Twitch[2], where most of the end user devices have very capable in terms of equipment. Beyond consume multimedia, user devices can adapt, cache [4] or relay popular video contents with idle computing, storage and network resources [5].

Ever since VoD and other network applications struggle with bandwidth and latency requirements, watching HD video (1080p) rather than video at a standard resolution (480p) typically increases the data traffic volume by around 4 times [6]. Many concepts proposes to shorten data transmission distance from source to destination, such as the concepts of fog computing [7], multi-access edge computing [8] and cloudlet [9]. All of them take into account cloud computing, but bring less powerful resources from far cloud deployments to closer to the network edge. At this manner, prevents the delay from the wired optical networks infrastructure and cloud response time.

The integration of cloud computing in HetNets multimedia scenarios virtually

[1] www.youtube.com
[2] www.twitch.tv

extends the mobile device resource capabilities and increases the Quality of Experience (QoE) [10]. More specifically, cloud computing provides a data center infrastructure with high computing resources accessible by mobile users across the Internet. However, thousands of users uploading/downloading multimedia content from the cloud quickly outstrip the bandwidth capacity and increase the delay [11]. On the other hand, fog computing considers cloud systems deployed closer to the users to meet their needs. Fog nodes might be deployed at different levels, from dedicated servers in the radio access or core network to the mobile devices themselves [7]. For instance, mobile devices could be a fog node that downloads the traffic from the cloud to share it locally via Device-to-Device (D2D) wireless communications [12]. However, the cooperation between fog and cloud computing must be seamless to bring important benefits for both user and network/content provider

Cloud computing provides a data center infrastructure with high computing resources accessible by mobile users across the Internet. However, thousands of users uploading/downloading multimedia content from the cloud quickly outstrip the bandwidth capacity and increase the delay [11]. Fog computing diverges from other edge computing concepts because it focuses on specific geographic regions [7]. The main characteristic of fog computing is its topology, i.e., the geographically distributed nodes that perform computation, offer storage and network services [13]. The basic elements of fog computing, called Fog Node (FN), can be designed and deployed in many fashions to cope with user needs, where fog nodes can be located in a logical and physical hierarchy, arranged in layers between the cloud on top and the mobile devices at the bottom [14]. Therefore, fog computing embraces a larger variety of applications at the same time. This is possible due to its expected resources to attend not only a specific edge application, but also multiples ones [15].

Mostly of data consumer devices request multimedia through wireless network. They rely on constrained energy banks, and occasionally offload computing processes to help them extend battery life. This case specially fits for smartphones, which carry numerous applications, own heterogeneous compute power and depend on odd wireless connection. Such scenario demands complex monitoring to meet low latency requirements and guarantee Quality of Service (QoS) for the network and QoE.

Multimedia application consumes a great amount of global traffic, more than wireless networks will deliver 50% by 2019 [16]. Most of it destined to energy-constrained mobile phone devices. Besides bring content closer to the end user, the video compression algorithms also evolved bringing the same amount of image quality with smaller video file size. Ever since release of new video coder-decoder (codec), such as H.265, video container size could reduce 40% compared to H.264, the current largely used codec, can drastically help to reduce multimedia traffic. However, only desktop processors adopted dedicated decoding instructions for H.265 decoding and few high-end smartphones started a couple of years ago.

Although H.265 [17] reduce video container size, it can significantly increase encoding and decoding processing. Therefore, mobile devices have to spend more energy to reproduce video content at same quality of H.264 and however spending requiring bytes [18]. Ever since multimedia content will represent 90% of global traffic by 2020 [19], network service providers and content provider have interest to mitigate multimedia traffic volume.

## 1.2 Motivation and challenges

The spreading of multimedia dissemination becomes a problem in an ossified network. The network may improve itself in order to comply with the growing data traffic demand of multimedia content [6]. However, fog computing can manage to improve QoE link quality since it can store content close to the end user preventing use of the full network path until the content provider [20]. Big multimedia content provider, such as Netflix [3] and YouTube, already apply content distribution caching on network service providers infrastructure, but yet not until the network edge probably gaining more efficiency.

Cloud computing brought many benefits, especially for content providers, since it empowers processing and rises storage volume at a lower cost [21]. Otherwise, maintain QoE requirements becomes more complex since it relies even more on the network infrastructure condition [22]. Fog computing proposes to improve QoS, especially on low latency requirements [23], but fog deployment still needs works on feasibility, performance analysis, achievable gains, among others.

In hot spots, such as in airports, a group of users located in the same area, sharing the same preferences or wireless network might search and watch hit videos, causing redundant transmissions [24]. For instance, studies [25], [26] showed that parts of traffic load are due to the download of some popular contents, *i.e.*, 10% of the top favorite videos account for nearly 80% of views. In this way, cache schemes available on FNs can reduce duplicated downloads, enabling users to request popular content from caches closer to the user, diminishing the need of user request directly to the multimedia service provider [27].

Distribute fog in multiple tiers until the edge and choosing a FN to store and transmit contents might require assessments gains. These FN, carry differences depending on tier level. Some of these FN can own abundant resources, such as a datacenter, while others are just a smartphone, designed for personal use. The resources in a FN can be computational power, storage volume, network link, and energy. FNs might range from dedicated servers in the core network to the mobile devices, and their capacities are growing exponentially by following the Moore Law [28].

Moreover, when considering the end user, to comply with QoE and QoS metrics miss user perception characteristics [2]. QoS can attend service level agreements of network provider but fails to attend user expectation. Video applications can struggle with stalls, bitrate switching, and playback start time not mapped by QoS metrics. QoE can be

[3] www.netflix.com

read by entities in the network and perform video adaptation to improve user experience and increase user browsing time on a web site.

Another challenge reside on the energy consumption related to video decoding in energy-constrained devices. A given mobile device may need more energy to reproduce a video encoded on H.265 [17]. Thus, an analysis of the difference between decoding energy consumption on smartphones become important since the energy expenditure during playtime can reduce the battery autonomy. For instance, the additional 40% [29], mostly pronounced at 1080p or higher resolutions, decoding complexity of H.265 can be proportional to the decoding energy consumption.

## 1.3 Objectives

This master thesis proposes a multi-tier fog architecture for VoD streaming. We seek to improve QoE of VoD streaming, such as reducing stalls and playback start time, and QoS, such as latency. We also expect to provide an architecture capable to deal with streams of video host services over distinct tiers in terms of localization and computational power and storage capacity. Moreover, we seek for flexibility for FN deployment over heterogeneous devices to leverage a collaborative environment. Finally, we seek to weight advantages for a newer video codec adoption and mobile devices inclusion as fog computing deployment device.

The architecture considers hosting a video content in different geographical areas for an experimental performance measurement of a fog computing tiers. The video player embedded to the smartphones collects QoE/QoS metrics and energy consumption during distinct network conditions and experimental scenarios. Each smartphone also reports the amount of energy consumed since codec and bitrates could differ between each other. Furthermore, the work collects real energy consumption on a physical network.

Thus, the objectives of this work include:

- Elaborate multi-tier fog computing architecture for video on demand streaming.
- Provide a collaborative environment for multiple tiers to improve QoE/QoS.
- Deploy the multi-tier fog computing architecture experimentally across a wide geographical area.
- Collect and analyze video service QoE/QoS metrics and energy consumption.
- Evaluate feasibility of the architecture deployed into tiers on real devices.

## 1.4 Contribution

This work has he following main contributions:

- A fog computing video on demand streaming architecture aware of QoE/QoS and energy constrained devices;
- A video player capable to return QoE/QoS and energy feedback;
- An assessment of how fog computing can improve QoE video streaming in terms of playback start time, freeze;
- An experimental validation of the proposed architecture.

## 1.5 Text Organization

The remaining of this document is structured as follows:

- Chapter 2: presents a deeper approach of the theoretical reference on cloud computing and fog computing. Explain the difference between codecs highlighting the reason to own distinct performances. The concepts behind QoE and QoS for the performance analysis.
- Chapter 3: presents state-of-the-art related works on fog computing and video on demand streaming evaluating advantages and disadvantages.
- Chapter 4: Details the proposed architecture.
- Chapter 5: Approaches the methodology to deploy the architecture on an experimental scenarios and explain the obtained results.
- Chapter 6: Concludes this work.

# CHAPTER 2

# Basic Concepts

This chapter presents the main concepts about video streaming, quality perceived by users and the paradigms of Cloud and Fog computing. Furthermore, we introduce topics related to the areas that this dissertation rely on.

## 2.1 Cloud Computing

Over the last decade, Cloud Computing paradigm gained prominence in both business and academia. This concept withstand the increasing processing of data that the Internet has generated all those years, especially after the easier access to the network around the world. In the business world, large corporations that maintain large data centers to offer their services to other companies to use as part of their infrastructure in a pay-per-use model. The customer pays for what they used within the cloud, thus obtaining flexibility and elasticity in the use of resources. However, in the academic world, many researches annually presented have the goal to of optimize platforms, concepts, services, access, energy consumption, allocation of resource, among others.

Cloud computing paradigm embraces distributed computing that achieved widespread adoption. It provides on-demand storage and processing capacity for high-capacity data centers and access over the Internet where these settings enable users to access any data types and/or any applications available on the Internet [21]. Three main hierarchical levels of cloud service are as follows:

- Infrastructure as a Service (IaaS): offer infrastructure to deploy server, storage and network.
- Platform as a Service (PaaS): offer a base system for deployment of applications such

as database management system without worry about physical devices maintenance.

- Software as a Service (SaaS): offers a higher level directed to end users who do not need to customize application configuration tuning performance tweaks.

Edge-centric Computing as an innovative paradigm which will put the computing of applications, data and services away from centralized nodes to the periphery of the network [30]. Data that would be executed centrally executed data within a cloud can be processed in a decentralized paradigm called Edge Computing [31], which consists of bringing the data for processing to the edge of the network. In this sense, fog computing can configure an instance of Edge Computing. Table 1 presents a comparison of features for deployment in the cloud and in the edge.

Table 1: Comparison between Cloud and Edge computing paradigms

| Requirements | Cloud Computing | Fog Computing |
|---|---|---|
| Latency | High | Low |
| Delay | High | Much Lower |
| Service localization | Far | Closer |
| Distance between Server and Client | Multiple hops | Few hops |
| Localization awareness | No | Yes |
| Geo-distribution | Centralized | Distributed |
| Mobility support | Limited | Supported |
| Real-time interaction | Supported | Supported |

## 2.2 Fog Computing

Connecting everyday physical objects to the Internet, enabling them to identify themselves to other devices and allow control and availability from anywhere, anyhow, and anytime in an intelligent manner demands a scalable infrastructure. Moreover, this connectivity describes an ecosystem driven by sensing, collection and exchange of information between smart devices running applications and services, as well as with the environment itself, with or without human intervention.

Fog computing concept proposes an extension of cloud computing paradigm at the edge of the network, thus allowing a new generation of applications and services [7], such as, connected industry, smart energy, connected building, smart retail, among others. In summary, fog own a set of features similar to a cloud but closer to users. These characteristics are listed below:

- Located in the edge of the network: provides smoothness to be aware of location and lower latency due to the proximity with the end users avoid some of the network issues;

- Geographically distributed: widely deployed over multiples micro datacenter or APs, owning limited resources in terms of processing, storage and communication capabilities;
- Large scale network sensors: provides computing and storage resources for many environment monitors easing the increased load over the cloud allowing increased scalability;
- High amount of nodes: attends geo-distributed sensors on smartcities enabling resilient and replicated services;
- Mobility support: allows direct communication to mobile devices migrate deployment nearing services considering users mobility;
- Real-time interaction: designed to provide low response time for critical applications complying with higher real-time requirements;
- Mostly wireless: driven to provide network services to wireless devices since most of them own energy constraints besides computational and storage limitation;
- Heterogeneity: capable to deal with a high diversity of sensors and actuator wireless devices;
- Interoperability: seamless support for several services across multiple domains due to high distributed deployment ;
- Analysis and data exchange support within the cloud: online analysis and interplay with the cloud.

Moreover, fog computing definition says that it is a highly virtualized platform typically provides computational, storage and network services between devices and traditional cloud computing data centers, but no exclusively located at the edge [7]. Many academic researches inherited many paradigms applied on cloud to fog computing, such as caching on a multi-tiered infrastructure, specially approached in this work.

## 2.3 Video codecs H.264 and H.265

Juurlink et al. [32] introduces H.264 codec. A video consists of a frame sequence quickly showed giving the illusion of motion. This frame rate typically stands between 20-30 frames per second. At these rates, even low-resolution images in long videos generates large files because mapping every pixel without compression demand enormous storage size for a video. Thus, codecs can encode the video sequence by compressing its size on disk also easing video stream due to reduced file size.

Video coding can exploit temporal and spatial redundancies. Spatial redundancy means that pixels spatially closer typically have approximate values. Temporal redundancy means that consecutive frames often have strong similarity by changing only the position of a single object through motion. This redundancy can mitigated by making the difference between one frame and another. Thus, a codec can store the next frame as only the difference of the previous one saving bits for its storage. This process, known as motion compensation and the motion vector, gives the distance of motion.

Currently, the most widely used standard in video applications is H.264 designed by the Motion Experts Group (MPEG), internally named as MPEG-4 part 10. This standard has improved the compression ratio of previous standards, such as H.262, MPEG-2, and MPEG-4 by a factor of two or more.

Wiegand et al. [33] presents H.264 standard fits to for high or low bitrate applications, high and low resolutions on a variety of networks and systems, such as streams on the Internet, disk storage and broadcast. ITU-T, Video Coding Experts Group (QoS), and ISO/IEC MPEG develop the video standard. The standard considers same block-based motion compensation by the encoding framework of the MPEG encoding standards and ITU-T video encoding [34]. It provides greater coding efficiency by adding new features that increase computational complexity.

Sullivan et al. [17] introduces the H.265 codec. The H.265 standard achieved a number of objectives, as well as greater coding efficiency, it aims to facilitate integration with data transmission systems, provide greater resiliency with data loss, and take advantage of parallel processing architectures. H.265 provides a data compression efficiency between 50% to 40% compared to H.264.

## 2.4 Quality of Service (QoS)

ITU defined QoS as a set of characteristics of a telecommunication service whose core is user satisfaction [35]. The Internet Engineering Task Force (IETF) summarizes QoS as a collection of requirements met during the transport of information from a particular service [36]. The parameters commonly used to measure QoS are width bandwidth, delay time between packets, variation of delay between packet loss index, and packet error probability.

For instance, a failure to transmit a video real-time to the device is shown in Figure 2. At some point in the display of the content, some regions of the image do not conform to the one transmitted by the server, shown on Figure 1. This may have happened for several factors, such as packet delays due to congestion network or packet loss due to wireless signal coverage.

Another example, when exist QoS degradation for video on demand, the server can retransmit the lost packets. However, end users can experience a high delay for playback start time, numerous stalls or, if the video distribution adapt to the network condition, annoying bitrate changes. Thus, QoS does not considers those streams behaviors and cannot attend video application only by detecting network conditions.

In a video streaming service, the measurement of QoS occurs in the instant that

Figure 1: Video transmitted by the server
Source: adapted from Maia [1]

Figure 2: Video received by the user
Source: adapted from Maia [1]

the server-generated packets transmit each video to the user device. This type of evaluation is called Network QoS (NQoS). On the other hand, one can investigate the relationship between the QoS parameters and the quality perceived by the user Perceived QoS (PQoS). Over the years, this term has evolved for QoE, where experience of the user is more valuable than the quality provided by the service.

## 2.5 Quality of Experience (QoE)

QoE is an assessment of user satisfaction with content displayed on the screen [37]. QoE is the degree of pleasure or annoyance of a person about an application, a service or a system [38]. Thus, QoE relates to the person's perception of the content displayed on the device. Perception comprises a processes of recognizing, organizing and understanding sensations from surround stimuli [39]. QoE bases the process of perception on the human audiovisual system, which relates the audition perception, such as volume moments of silence or audio distortion, or visual of color, such as variation of light intensity or failure in some pixel, of the user about content displayed to it on some device.

The process of perception begins in the contextual information related to stimuli originated from the environment where the person locates. In addition, relates to the kind of content and the noises around, thus their senses. The brain convert the stimuli into neural representations, mental and physical states of the person, and with the assumptions, such attitudes and concepts. Hence, awareness of quality in action for the person concentrate in some type of evaluation of quality. The result allows the comparison with previous cases to, finally, evaluate positively or negatively the quality. Finally, based on this evaluation and conditions of his state and his assumptions, the person opinion about quality [38].

In the beginning, the evaluation of the QoE considered only subjective parameters, such as the user's perception of the content displayed on the device measured by words such as excellent, good, acceptable, bad or terrible. In addition, other parameters taken into account, such as cost, availability, usability and fidelity [40]. However, due to the difficult to subjective evaluation of videos, hybrid and objective strategies emerged and we focus on objective strategies depicted on Figure 3. The common objective QoE metrics are as follows:

- Playback Start Time: the playback start time or the initial delay consists of the time duration before a video starts to playout. The playback start time typically starts from the time taken to download the HyperText Markup Language (HTML) page (or manifest file), load the video player dependencies, and to finally playback the initial part of the video. In the case of streaming videos, the player starts playing the video only after the download of part of the file, known as Initial Buffering. This fist buffering activity allows the player to overcome the effect of the delay jitter incurred during the data transfer on the video playback. QoE can be affected due to higher initial push data rates and latter reduction of certain VoD services [41], which rise the playback start time. It was observed in [42] that playback start time had a significant influence on user retainment and extending more than 2 seconds results in the viewer abandoning the video.

- Freeze times: an interruption occurs when the playback of the video temporarily stalls. A streaming player downloads the initial parts of the multimedia content into a playout buffer before the video start playing and it continues playing as long as the playback rate be equal or higher than the video bitrate. This happens because the buffer depletion and the video player has to wait to be filled leading to poor QoE [43]. These interruption events also can be referred as re-buffering and the frequency buffering events. Users who experienced more during the video playback tend to watch the video for shorter durations [42] and probably become dissatisfied in the case of four or more interruptions for videos [44]. Hence, freeze times during the video playback is an important metric to measure the satisfaction of the users [22].

- Duration of freeze times: apart from the number of times the playback interruptions, the duration is also an important QoE metric. If the interruption duration is one second, the users has lower dissatisfaction when compared to 3 seconds of interruption while watching YouTube videos [43], [44]. However, viewers prefer a single long stall than many short stall events [45]. Hence, beyond the number of interruptions, the duration of each interruptions also causes distinct effects on the QoE.

- Quality of the Video File: the quality of a video stream mostly depends on the encoding rate. The Encoding rate is the average data required to play one second of the video, also referred as video bitrate. The video bitrate affects the QoE of the users [46]. A high quality HD (HD) video might require a more data for each frame and hence, results in a higher bitrate. Progressive download streams typically stick to the same bitrate throughout the duration of the playback, irrespective of the change in the network quality. Adaptive streaming techniques, like real-time streaming and HABS, vary the encoding rate depending on the network parameters. There are other video characteristics that have been used to represent the quality of the video such as the contrast, blurring [47], and blockiness [48], [49], [50], [51] Blockiness manifests as a block appearing in the video. It is caused by the block-based coding schemes such as H.261, H.263, and MPEG-1.

- Bitrate Switching Events: the bitrate switching events relates to the HTTP Based Adaptive Streaming technique (HABS). For Dynamic Adaptive Streaming over HTTP (DASH) videos, the player tends to pick a lower initial startup and gradually keeps increasing the quality before settling at a suitable bitrate. A degrade network condition can cause a bitrate reduction minimizing interruptions during the playback but can increase again if the network condition improve. However, frequent switching in bit rates can degrade the users QoE [52]. Hence, video bitrate switch might be reduced in order to achieve an average bitrate preventing worsen QoE [52], [46].

- User Engagement: user Engagement reflects the user involvement and interaction with the video. User engagement measures in terms of the number of views and the playtime of The video. However, the playtime might not reflect the actual amount of time the user spends watching the video without any distraction. Quantify the user?s focus is a subjective metric, thus difficult to map. The users who are satisfied with the content and the QoE of the streaming session tend to spend more time watching the video [53].

## 2.6 Chapter Conclusions

This chapter introduced knowledge about cloud computing, a paradigm to centralize processing and storage in a highly scalable fashion. In addition, it introduced fog computing features and how to overcome some issues of the cloud. Furthermore,

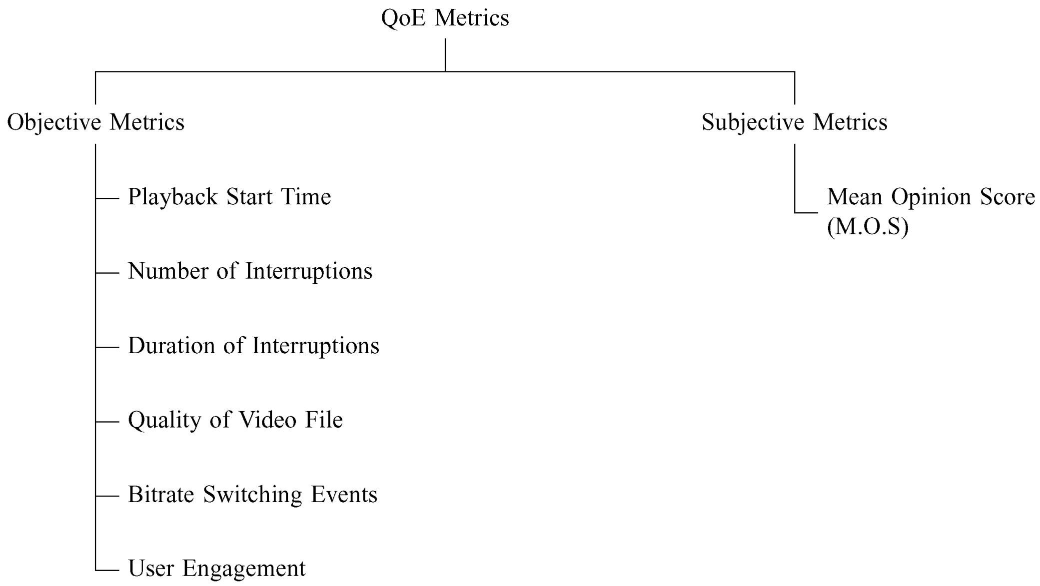


Figure 3: Opinion score formation process
Source: adapted from Juluri et al. [2]

it explained some quality measurement methods for network and applications in order to improve user experience. Finally, it explained basic video codecs improvements of compression of videos and the file size reduction of two generations of codecs.

# CHAPTER 3

# Related Works

This chapter introduces related works of this dissertation highlighting the state-of-art and missing improvements considering fog computing paradigm for multimedia distribution and D2D communication. The architecture designs directly related to fog computing paradigm for low latency applications. Content dissemination focus on reduce redundancy of data transmission on edge nodes. D2D communication focus in optimizations between end user devices.

## 3.1 Architecture Designs

Li et al. [54] propose a two-level hierarchical Fog interconnection along Cloud-User continuum consisting of Fog servers (top level) and Fog edge nodes (bottom level) in order to consolidate data communication between end devices and the Cloud. They focus on data-centered fog platform to support smart living alongside a dataflow analysis on an Information-centric fog-to-fog architecture. The design is thus limited to cache not considering special treatment to multimedia applications.

Nguyen et al. [55] introduced ICN-Fog, a novel horizontal Fog-to-Fog layer enabled by ICN. They suggest an ICN-Fog which improves applications with horizontal data transfer in the Fog layer, distributed processing among FNs, and built-in mobility support thanks to the smart connectionless name-based Fog-to-Fog data communications. They include an energy management system since most of FNs depend on battery, a finite resource which welcomes every energy saving.

Zhang et al. [56] proposed a regional cooperative fog-computing-based intelligent vehicular network architecture for dealing with big Internet of Vehicles data in the smart city. They discussed mobility control, multi-source data acquisition, distributed computation and storage, and multi-path data transmission. Moreover, they presented a hierarchical model with intra-fog and inter-fog resource management, and energy efficiency, and packet dropping rates. They considered energy-aware and QoS-aware resource management. However did not consider video transmission neither QoE.

Vilalta et al. [57] proposed an architecture for secure, highly distributed, and ultra-dense fog computing infrastructure, which can be allocated at the extreme edge of a wired/wireless network for a telecom operator for third parties (e.g., smart cities, vertical industries, and IoT). They suggested strengthening the position of the mobile network and cloud markets by scaling, control, and disposing services on FNs. However, they focused on concept for IoT services.

## 3.2 Content Dissemination

Ahlehagh and Dey [58] introduced a distributed caching scheme of videos at the Base Stations (BS) of the Radio Access Network (RAN) to reduce backhaul transmission, thus improving QoE and increasing the number of simultaneous video requests. It proposes a caching architecture with a large number of micro-caches storing hundreds compared to a CDN, which stores millions of videos. However, they proposed caching policies in order to avoid cache misses of videos needed to be fetched from Internet CDNs. Moreover, they developed a discrete event statistical simulation framework for RAN caching performance evaluation. They state an improvement to attend initial delay requirements by almost 60% and the number of concurrent video requests raise up to 100%. In addition, their user preference profile cache policy can enhance network capacity by up to 30% compared to others regular caching policies.

Gomes et al. [59] considered some cache schemes for the LTE network edge with mobile user by bringing content stored in the Information Centric Networking (ICN) to the LTE edge. This work considered cache for video content categorized based on its popularity. They propose cache schemes where contents follow possibly future interested users towards them. Moreover, they state a faster delivery to end users and more efficiently storage resources management at the core of the network.

Bozorgchenani et al. [60] suggests a fog computing infrastructure with two layers: one including FNs and another the fog Access Points (AP). They define FNs as battery operated and the fog AP as connected to the electrical networks having unlimited energy. Moreover, fog AP can offload computational tasks from FNs due to their higher storage and computational capabilities. They demonstrated with simulation how partial offloading had a profound impact on the network lifetime and reduced energy consumption and task processing delay.

Gao et al. [61] suggest reduce Fog-Cloud communication through disruption tolerable network techniques utilizing storage on moving end devices and/or moving FNs. Moreover, they suggest precaching contents on FNs in order to provide high quality with lower cost for data distributions. However, they simulated their proposal and state it does not fit in urgent applications neither high volume contents. However, their proposal does not work for low latency content dissemination but showed efficient success ratio for data-dissemination.

## 3.3 D2D

Jo et al. [12] introduced a Content Delivery Network (CDN) for mobile devices in a heterogeneous Radio Access Network (RAN) architecture in order to enhance the throughput by using more than one network interfaces at the same time. They propose that some of the devices act as relay node to forward the data for disconnected devices. This paper regards on dense network scenario, such as happen in airports, stadiums, and train station, where a group of mobile devices could share the WiFi to connect them. However, such work does not consider caching schemes, transcoding, neither QoE to improve the network performance.

Wang and Lin [62] introduced a network paradigm by taking advantage of an emerging type of user-provided network. They propose the use of intermediate nodes acting as relays, increasing the network range and providing better network connectivity for mobile user. This brings benefits for scenarios with high signal attenuation caused by obstacles, for example, building blocking the signal from the BS for an entire group of mobile devices. Moreover, they suggested a scheme where end user plays important role evaluate QoE and improve the network.

Karvounas et al. [63] considered a D2D communication to solve persistent issues of mobile network. They suggested an AP to expand the coverage of the infrastructure or even opportunistically increase network capacity when an AP faces congestion issues. They experimentally introduced a real-time video transmission via D2D with multiple hops, and considered a mesh network protocol for the D2D communication. They observed that each hop increases the delay and reduces the throughput.

Wang et al. [64] evaluated cache strategies at the Evolved Packet Core (EPC), RAN and using D2D communication of 5G mobile networks. They suggest caching at intermediate servers, such as middleboxes, gateways, or routers, easily reducing duplicated transmissions, hence significantly eliminating redundant traffic. They simulated and evaluated the performance of the proposed scheme and concluded it as a new relevant opportunity.

## 3.4 Chapter Conclusions

Based on the analysis of our related work, we conclude that it is essential to evaluate the performance of cache strategies deployed at different levels of the network infrastructure beyond the core and network edge, where caching the content closer to the network edge improves the QoE. In addition, we can join the state-of-the-art and propose an architecture to integrate most of it and add video adaptation awareness, since multimedia represents the biggest data traffic in the current network.

Table 2: Related works

| Proposal | Caching | Content Adaptation | Network | Energy | Offload |
|---|---|---|---|---|---|
| Li et al. [54] | yes | no | CDN | no | no |
| Nguyen et al. [55] | yes | no | ICN | yes | no |
| Zhang et al. [56] | no | no | Standard | yes | no |
| Vilalta et al. [57] | no | no | Standard | no | no |
| Ahlehagh and Dey [58] | yes | no | CDN | no | no |
| Gomes et al. [59] | yes | no | ICN | no | no |
| Bozorgchenani et al. [60] | no | no | Standard | yes | yes |
| Gao et al. [61] | yes | no | Standard | no | no |
| Jo et al. [12] | no | no | CDN | no | yes |
| Wang and Lin [62] | no | no | Standard | no | yes |
| Karvounas et al. [63] | no | no | Standard | no | yes |
| Wang et al. [64] | yes | no | Standard | no | yes |
| **Current proposal** | yes | yes | Standard | yes | yes |

# CHAPTER 4

# A Multi-tier Fog Architecture for Video on Demand Streaming

This chapter defines the proposed multi-tier fog architecture designed to optimize video on demand streaming on wired and wireless networks towards mobile devices. The elements of the architecture intends to improve video dissemination from the cloud, infrastructure of network operators, as well as mobile devices.

## 4.1 Scenario

We specify the scenario as, the video service provider send multimedia content to the client application. A centralized controller in the cloud might respond each client request and orchestrate/control available network elements. In the case of degraded QoE, the controller instantiates FNs in a feasible fashion to improve content distribution through caching schemes, video adaptation and/or relaying. These FN fit into groups according to their features and network topology.

We consider three Tiers depicted in Figure 4. Tier 1 correspond to the cloud responsible for orchestration and control of the multi-tier fog architecture, including lower fog tiers. This tier has the Service Provider, which host all videos in a Video Host Service platform, and the Fog Controller Host to orchestrate and decide what, where, and when the multimedia content must be transcoded, cached or relayed according to a specific algorithms. This host executes Fog Application, responsible for orchestration and control, and Media Server Application intended for transcoding and caching in the cloud.

Tier 2 correspond network infrastructure of Private Network Operator or Mobile Network Operator can host a Media Server Application. The First operator consist Inter-

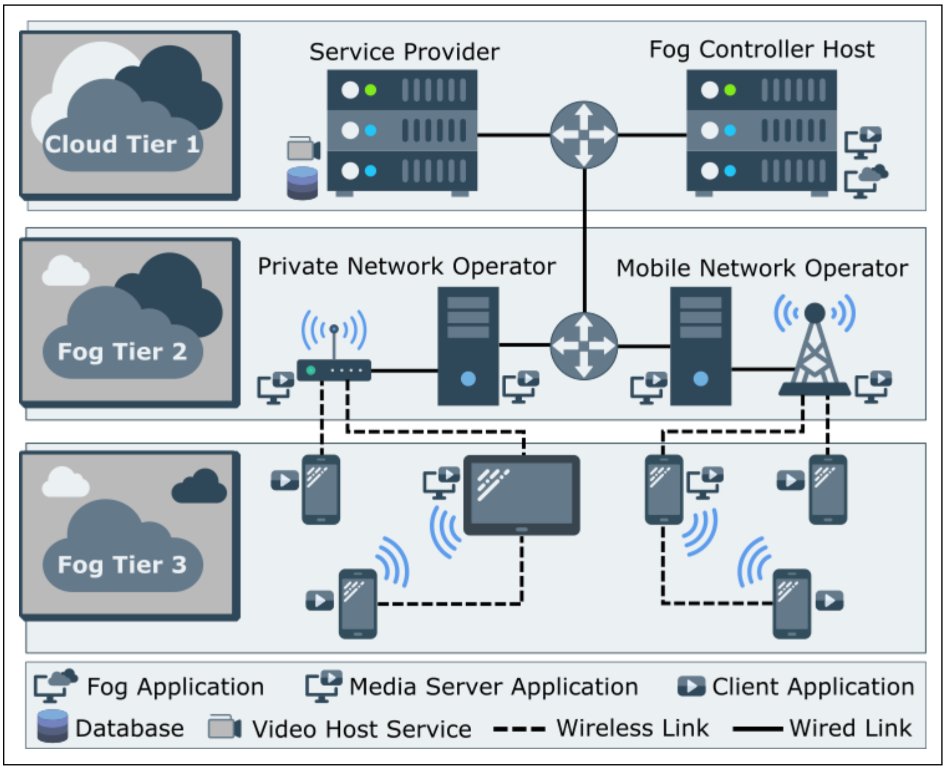


Figure 4: Multi-tier Fog Computing Scenario

net Access for dense wireless mobile networks with short-range interfaces, such as WiFi or Bluetooth, where APs or its datacenter unit can work as a FN. The second operator consist of mobile network with higher range and more user. It also can work as a FN hosting Media Server Application on its core or the edge elements, such as EPC and BS units.

Tier 3 deal with mobile devices considering their constraints, such as reduced computational power, storage and energy availability. However, these devices can contribute to network offloading if they share their resources to cache and/or relay popular content in dense areas. In addition, they execute the Client Application for decoding videos in high quality demanding high bandwidth from wired and wireless links.

## 4.2 Architecture

This section introduces the proposed multi-tier fog architecture. The architecture considers centralized cloud computing together with distributed FNs, providing cloud services for multimedia distribution client applications in a collaborative fashion. Cloud execute the Fog Application, while FNs execute Media Server and Client Applications. The centrally managed FNs perform data processing functions, retransmission, caching, and requests/displays the content, since user device also can turn to a FN.

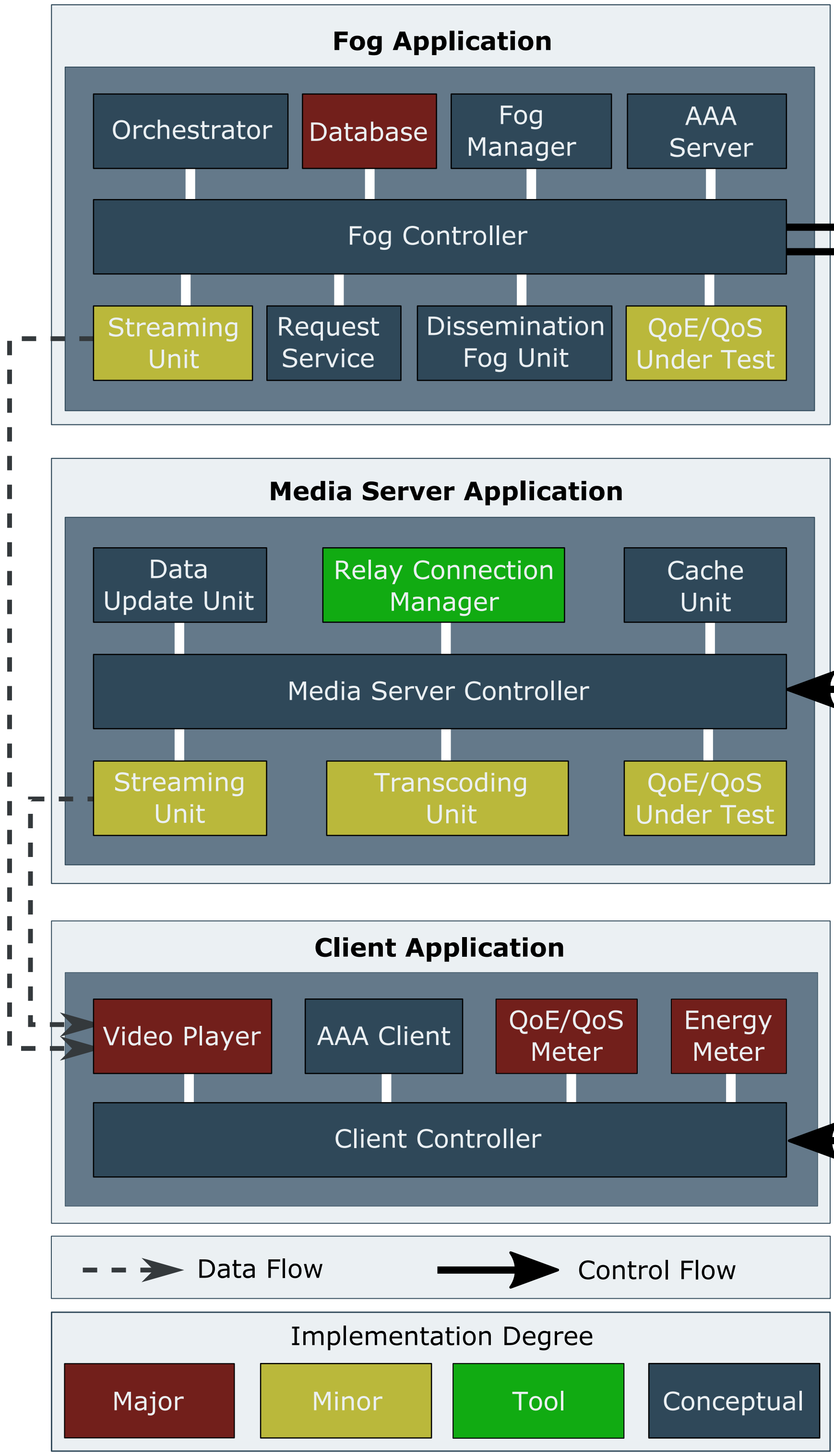


Figure 5: Multi-tier fog architecture

### 4.2.1 Fog Application

This application run strictly on Tier 1 and we specify the following modules: Fog Controller, Orchestrator, Database, Fog Manager, AAA Server, Streaming Unit, Dissemination Fog Unit, Request Service, and QoE/QoS Under Test. The Orchestrator conducts any type of management and decision making, *i.e.*, it orchestrates the communications between all network elements, decides about fog deployment, and considers specific algorithms to decide where, what, and when the multimedia content must be cached and/or transcoded. Orchestrator considers input from the QoE/QoS Under Test regarding the current network state and user experience, and operator specific information, such as network policies or service level agreements. The Fog Controller manages/provides communication among internal modules, synchronizes control and data flow exchange, and sends decisions taken by the Orchestrator to Media Server, and/or Client Controllers.

### 4.2.2 Media Server Application

In the case of poor QoE during the video streaming, FNs might also run Media Server Application in the radio access or core network. More specifically, the Fog Controller deploys a Media Server Application in the BS or the AP. Media Server Controller connects the modules Data Update Unit, Relay Connection Manager, Cache Unit, Streaming Unit, Transcoding Unit, and QoE/QoS Under Test in any Tier. However, considering Tier 2 and 3 elements, Transcoding Unit and Relay Connection Manager may not often execute on Tiers 3 and 2, respectively. The first because of energy constraints and the second because of exclusivity of D2D in Tier 3. The Media Server Controller plays the role of an interface between Media Server and Fog Applications synchronize the control, and data flow exchange in both directions, manage/provide communication among internal modules, process requests from the Fog Controller, and apply any decision received from the Fog Controller.

In Tier 2, a given network element becomes a FN when it adapts and/or stores content avoiding the connected mobile devices to request data from the degraded network path and reduce transmission redundancy. In more details, a FN host the Media Server Application assigned to execute some of its modules for transcoding the video codec, bitrate, or resolution according to the network conditions, device capabilities, or QoE [65] and/or caching to reduce transmission redundancy. From this, the Streaming Unit distribute content closer to the user device besides collecting QoE/QoS data preventing it transmission until the Tier 1 and reducing link load.

In Tier 3, a given mobile device becomes a FN when it download and share the content among its neighboring mobile devices using Wi-Fi direct, Bluetooth, or another D2D wireless technology. More specifically, the Client Controller receives an assignment from the Fog Controller to become a FN for relay, and it starts the Streaming Unit and the Relay Connection Manager. The Relay Connection Manager allows the mobile device to relay the video content via D2D communication, to discover mobile device neighbors with similar needs, and to manage the existing D2D connections. Finally, the Streaming Unit enables the mobile device to stream the video content to neighbor mobile devices, such as provided by HTTP server application.

### 4.2.3 Client Application

The Client Application considers a Client Controller that communicates with four modules, namely Video Player, Authentication, Authorization, and Accounting (AAA) Client, QoE/QoS Meter and the Energy Meter. The Client Controller has the role to interface control flow between Fog and Client Applications, synchronizing data flow exchange in both directions. This controller manages/provides communication among internal modules, processes requests and applies decisions received from the Fog Controller. The AAA Client provides the authentication of mobile devices and users, the authorization for consumption of networking and computational resources, and the accounting for collection of relevant data. Hence, AAA Client fulfills the security requirements. The Energy Meter collects energy consumption of the mobile device.

The Video Player downloads the video content from the Streaming Unit located in a given FN and shows the content to the user. More specifically, the orchestrator decides the Streaming Unit for downloading, and the Fog Controller sends this information to the Client Controller. The architecture support existing video players on mobile devices, such as ffplay. During video content displaying, the QoE/QoS Meter collects QoE and QoS measurements using a player developed for this work. The player enables to collect QoS metric or QoE metrics (e.g., playback start time, duration of freezes, and Mean Opinion Score (MOS)). Therefore, the Orchestrator need those QoE/QoS measurements to take decisions instantiating Media Server Application into some Tier to meet user needs.

## 4.3 Modules Description

This section describes more deeply the modules of the architecture, highlighting concepts behind each module and implemented mechanisms for testing and validation on the experimental environment.

### 4.3.1 Video Player

Responsible to play multimedia contents into distinct encoding parameters. These contents codification can be into multiple codecs and packed together in a container. The video player reads the file stored in a container kind, such as MP4, MKV, AVI, among others, and multimedia encoding characteristics, such as bitrate, frames per second, video and audio codecs, constant or variable bitrates, among others. The player might have support to play videos distributed over the network to receive data with HTTP protocol.

The architecture proposes to improve video QoE by approximating multimedia content closer to the user. In this sense, firstly the Fog Application require QoE/QoS collected from mobile devices for decision-making. The video player module own the best standpoint because it can internally register QoE/QoS degradation events, such as playback start time, number and duration time of freezes while playing the video.

Therefore, we customized ijkplayer [66] based on [67] and embedded QoE/QoS Meter and an Energy Meter.

### 4.3.2 QoE/QoS Meter

The first feature gathers information about how much time takes to the user click on the screen and has to wait for playing. The QoE metric playback start time maps this condition allowing optimization in the video dissemination. Users normally stop to watch video when it takes high buffering time potentially harming Service Provider interesting because users spend less time into the platform. The video playback start depends on main four events, video request, transmission start, buffering and the video playing.

A given client send a request for a video to the media server, which respond by send video frames to the client. However, the video starts to play only when the video player detects enough frames on its buffer causing a delay for the video playback start time. Video bit rate and link quality can worse this metric because higher bit rates demand a bigger amount of data in the buffer and the link quality can worse even more. Figure 6 depicts this process.

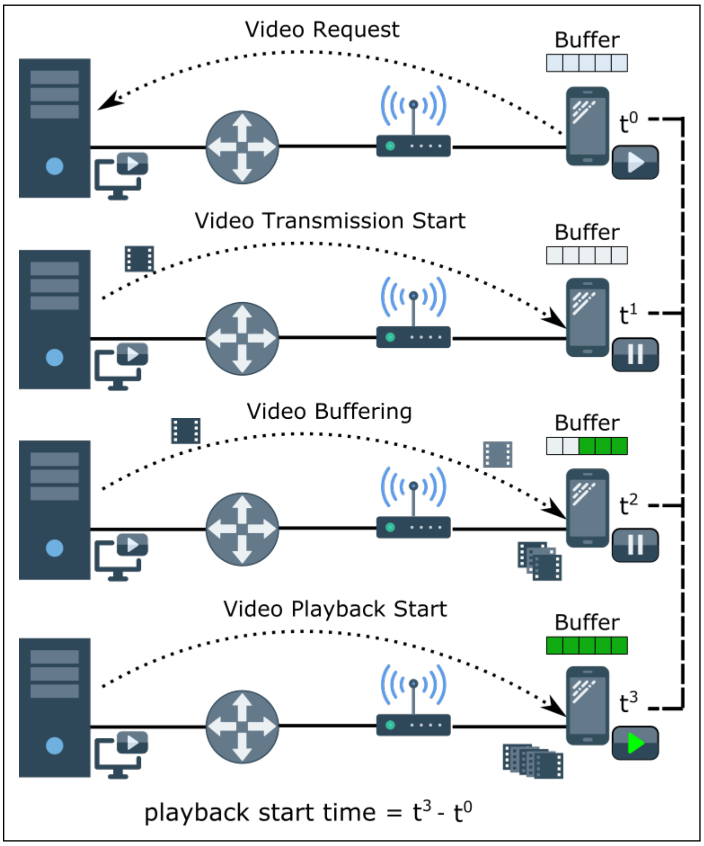


Figure 6: Video playback start time diagram

The second feature gets the number of freeze times while the video plays. Users prefer bigger freeze times than many small freeze times because it ruins the continuousness of the video flow. When the bandwidth available cannot meet the video bit rate, the player stops to play waiting for re-buffering. However, ever since the video can automatically play small portions of the video, such as one second at a time with half-second freeze. Summing all the video froze, user could have a better QoE if the player waited for more frames into buffer even if it had to wait for continuous time, such as 5, 10, or even more seconds.

The third feature gets how much time the player paused the video for re-buffering. This event can happen more or less frequently but this metric sum the time spent during all re-buffering events after completely play the video. It matters to QoE because the media server can optimize QoE even if it cannot reduce the video froze but control the player to only play the video after being capable to reproduce a minimum time without pause it again. Figure 7 depicts this process.

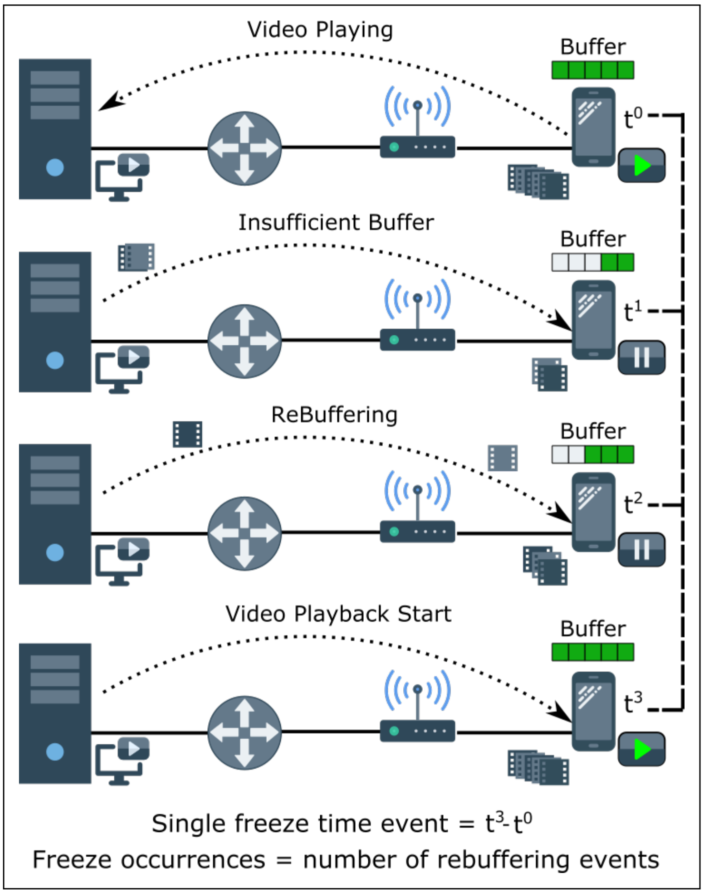


Figure 7: Video playback start time diagram

Besides getting QoE metrics, the video player also performs an active QoS test when a video requisition occurs. Before start buffering, the player checks QoS with Round-Time Trip (RTT) between the Video Player and the Streaming Unit host. This QoS metric measures the length of time of the round time trip of a packet between source and destination. The orchestrator can set the video player to request video from a different host with better QoS in order to improve user experience.

### 4.3.3 Energy Meter

The energy consumption feature seeks to monitor the amount of energy to watch each video. Every time the smartphone request a video, the energy meter registers the battery percentage and, when the playback finishes, it registers the remaining battery percentage. The player keeps track of energy consumption in order to attain codec differences regarding decoding video from lower to higher bit rates. A given user may prefer worse image quality than spend more energy from a device with low battery.

The Android operational system separates energy consumption per application or even shows the current consumption from battery at real-time. However, not every smartphone returns this information because it demands harder kernel implementations, depends on the current battery state varying on new or old ones and demands more energy for monitoring itself since an app must be running to register more precisely. Therefore, checking only battery percentage can return worthy information without significant imprecisions.

### 4.3.4 Streaming Unit

We deployed the streaming unit with ffmpeg [68] because it can decode, encode, transcode, among others things, many of leading multimedia technologies. It runs in a wide variety of operational systems, machine architectures and configurations. It is one of the leading multimedia frameworks capable to reproduce most of the standards, from committees, community or even corporations. Ffmpeg combines the best free software options available perfectly fitting the needs of this dissertation.

Ffmpeg with the broadcasting server, named ffserver, can distribute videos to multiple clients. It has the feature to separate the streaming system into pieces deployed in other devices around the world. This feature allows distributing the processing of the streaming media system depicted by 8. Ffserver works with four elements, which are input sources, feeds, streams and media player.

The input source send video stream to ffserver to centralize the video distribution in a single IP address. The feeds allow the distribution of the same content in different output format(i.e., in 2250 kbps bitrate or 1080p resolution) at the same time. Streams represents a connection point for viewers watch a specific stream and can handle with multiple clients. Then, adversely of feeds, streams concede the opportunity to fit the content to the end device needs. Finally, the media player just display the video content on the screen of the device.

For this, we deployed ffserver as base server framework. Moreover, it supports both codecs that we evaluate on this work being one of the most optimized platform for video stream, adaptation and playback. We set a ffmpeg configuration file to stream the video contents in a remote server.

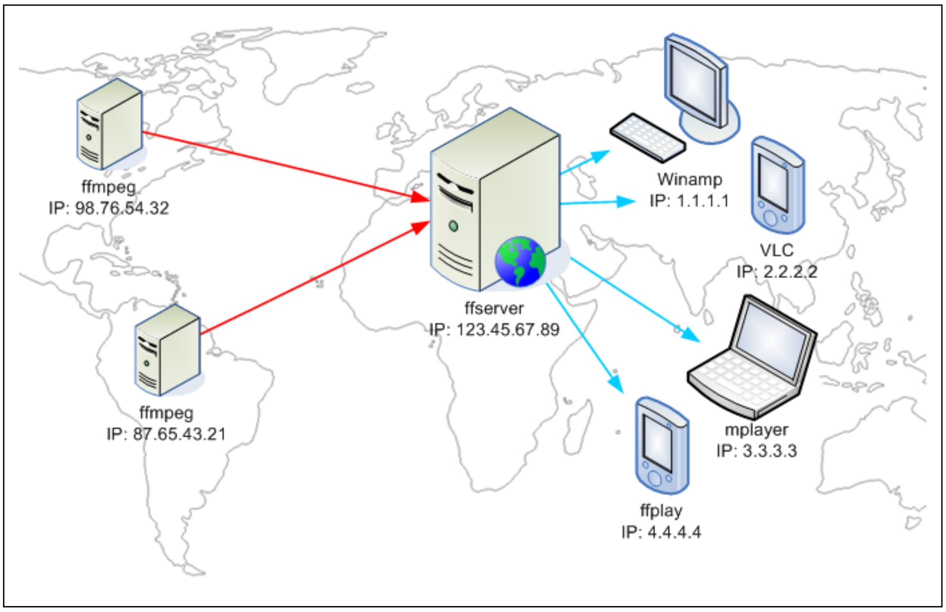


Figure 8: Video playback start time diagram
Source: Adapted from ffserver [3]

### 4.3.5 Database

The database stores the video contents regarding some encoding parameters. The database always own the video content in the best visual quality at a price of consume a bigger storage volume. Sometimes, the database also stores the video content into others encoded parameters avoid encoding redundancy for every content request for each client, as done by YouTube[1] and Netflix[2]. The database can feed directly the video player and the Media Server Application. This application holds a fraction of the database contents

For the database, we gathered a video coded in H.264 approximately at 10.000 kbps and re-encoded the videos at the same bitrates of 2250, 1000 and 400 kbps on both codecs. Moreover, the video resolution had 540p, 360p and 180p, respectively, according to Microsoft Azure presets [3]. Azure presets does not consider video stream with H.265, however we tried to encode the H.265 videos considering equals parameters as possible. Therefore, the video container had the same resolution, frame rate, bitrate, among others parameters.

### 4.3.6 QoE/QoS Under Test

Every smartphone generates a log file containing the collect QoE and QoS events, which can be upload to the Media Server Application host containing the current Streaming Unit in order to trigger actions of the Orchestrator. However, this dissertation used those data only as proof-of-concept to point gains automating this orchestration process

[1]www.youtube.com
[2]www.netflix.com
[3]https://docs.microsoft.com/en-us/azure/media-services/media-services-mes-presets-overview

with QoE and QoS inputs.

### 4.3.7 Relay Connection Manager

The relay connection manager enables stream communication between two mobile devices for content distribution. Both devices can discover and connect to each other using HTTP and Websockets providing a simple service establishing. Normally, the smartphone will not store the content for relaying because of small storage volume availability. Therefore, the device works just as a relay FN. Another use case happens when the smartphone previously caches due to high content popularity.

In addition, the relay devices can sense their communication link. They read QoS, such as bandwidth, packet loss, or latency and signal power. These metrics can help the orchestrator to discover needs for deployment the relay in a new FN, keeping the current solution and finish when not needed anymore. Finally, the rewarding system might monitor how collaborative the end user did to give a fair incentive for such.

### 4.3.8 Transcoding Unit

A video stream adaptation can occur in two fashions. One of them adapts the video in real-time by reducing bitrate and others parameters. The other way by encoding the content into multiple sets avoiding to constantly or occasionally processor units for this task. Due to storage cost reducing over the last years, big video host websites use the second way, however caching all contents version along the multi-tiers cause lower cache efficiency. Transcoding can adapt a popular video content version for transmission over a degraded link not capable to receive a content at the chosen cached version.

Besides changing the video codec, bitrate and resolution sets, the transcoding unit can also adapt itself in order to fulfill the current processing load. Apart from the three previously cited settings, it can change preset. Presets keep a collection of options regarding encoding speed to compression ratio. Fast presets sacrifice video quality in order to dispose faster a content. Therefore, depending on the FN transcoding load, these presets can be changed in order adapt the processing capacity of a given FN.

### 4.3.9 Cache Unit

Stores popular content in a storage. This unit refreshes the stored data following some policies according to the data popularity and hit rate. The caching varies the storage volume depending on the deployed device on each Tier. Therefore, Device closer to Tier 1 have more capacity than the ones on the last Tier. This happens because user devices normally have unit of gigabytes in their storage capacity specific for personal purposes. However, Cloud and the mid-layer serve a high amount of user devices and become more suitable to embed higher storage capacity.

### 4.3.10 Data Update Unit

Responsible to monitor and manage the data stored in the Cache Unit applying the designed policies to fulfill the available caching storage room. This Unit detect ascending and descending content popularity and run replacement algorithms to determine content replacement. This algorithm can prefetch data in advance when expect a future high demand for a content or a set of contents. In summary, the Data Update Unit deals with the life cycle of the whole contents deployed in a specific FN also updating whenever a content receive a newer version in the original server.

### 4.3.11 Dissemination Unit

A large deployment number of FNs can occur and these nodes may have to forward data across multiple hops to deliver a data. Each FN might have an identification in order to be achievable by the other FNs. This network knowledge allows easier cooperation between the devices giving access to more resources in the FN. FNs might join resources in order to offer higher storage, computational and network capacity to grant better QoE and QoS. At this fashion, the dissemination unit forward contents to the designed collaborative cache unit of different FNs widen the power of fog attending a region.

### 4.3.12 Request Service

When the video player request a video content, the video downloads from the cloud or another FN. The video player start requesting from the cloud, however the request service redirect the connection to the most suitable FN to send data. The request service establishes the connection and can live change the requesting address in cases of degradation of the link or if a better FN to forward video content.

### 4.3.13 AAA Server

The access between FNs and network devices will facilitate communication between all devices enlarging security issues. In addition, the wireless network allowed a spreading of mobile devices increasing complexity and demand on AAA. Therefore, an AAA server might enable mobility and dynamic security for devices of Tier 3. The tasks of the AAA server are Authentication to identify a device, Authorization to allow the actions of a given device and Accounting in order to register the actions executed in the fog.

The AAA server has with five components, Client, Authenticator, policy information, policy decision and reporting system. The client attempts to access the network and can operate as a FN. The authenticator enforces the terms of the access for each client. The policy information maintains a database of access decision for each client or

FN group. The policy decision makes final decisions around the fog access. Finally, the report system tracks the client actions keeping an historic serving as input for a billing and reward mechanism.

### 4.3.14 Fog Manager

The available devices for fog deployment might have custom configurations and be periodically tracked. When a demand for fog emerges, the Fog Manager performs as discovery device in order to find the most efficient nodes. Moreover, the fog manager also update configurations settings inside each device, as well as keep mobility tracking. In addition, the devices will communicate with each other intermediated by fog manager to grant a higher security level.

### 4.3.15 Orchestrator

The proposed architecture considers a few services to enable better video content dissemination. These services run on demand, thus the orchestrator manages life-cycle management of services instantiation, scale-out/in and termination. Moreover, a wide knowledge of the whole network reached by the device mobility and capacity monitoring allows the orchestrator to make decisions for network services improvements.

The Orchestrator can consider information about QoE, QoS, topology, video Content, operator, and others, for decision-making. The orchestrator might run specific algorithms deciding where and when to deploy FNs that run cloud services for video distribution in the case of poor user experience. In this context, network machine learning could improve the orchestrator capabilities. In addition, the orchestrator must deal with users moving between different locations quite often in the current HetNets scenarios, which hamper the delivery of videos with QoE support

## 4.4 Chapter Conclusions

We highlight the elements implementation degree regarding code intervention in open-source softwares implemented for test and validation on an experimental environment. In addition, we also used free and open-source tools with custom configuration file settings to obtain a usable experimental environment with reasonable control over it.

We consider four categories of implementation degree: Conceptual, Tool, Minor and Major. Conceptual when we only describe attributions of an element, Tool when we used a software of simples use to create the experimental environment, Minor for more elaborated configurations settings and Major for coding interventions or creation. Table 3 depicts implementation degree of the elements.

In summary, the role of all modules are as follows:

Table 3: Modules Implementation Degree

| Application | Module | Degree |
|---|---|---|
| Fog | Fog Controller | Conceptual |
| | Orchestrator | Conceptual |
| | Database | Major |
| | Fog Manager | Conceptual |
| | AAA Server | Conceptual |
| | Streaming Unit | Minor |
| | Request Service | Conceptual |
| | Dissemination Unit | Conceptual |
| | QoE/QoS Under Test | Minor |
| Media Server | Media Server Controller | Conceptual |
| | Data Update Unit | Conceptual |
| | Relay Connection Manager | Tool |
| | Cache Unit | Conceptual |
| | Streaming Unit | Minor |
| | Transcoding Unit | Minor |
| | QoE/QoS Under Test | Minor |
| Client | Client Controller | Conceptual |
| | Video Player | Major |
| | AAA Client | Conceptual |
| | QoE/QoS Meter | Major |
| | Energy Mter | Major |

- Fog Controller: synchronizes control signaling from and to the others controllers;
- Orchestrator: performs decision-making and management;
- Database: store the videos;
- Fog Manager: leads instantiated FNs;
- AAA Server: perform security measures, such as auditing and billing;
- Streaming Unit: distributes multimedia content;
- Request Service: indicates from which FN the Video Player must download the content;
- Dissemination Unit: distributes the multimedia content in different FNs;
- QoE/QoS Under Test: collects QoE/QoS measurements feedback from user devices.
- Media Server Controller: receives a role in adapting a given video and it starts the Transcoding Unit to apply for such role;
- Data Update Unit: maintains updated the content copies stored in the Cache Unit

- Relay Connection Manager: set D2D communication configuration in mobile devices for data exchange;
- Cache Unit: stores redundant copies of a given content near to the user;
- Transcoding Unit: adapts the video codec, bit rate, or resolution according to the network conditions, device capabilities, or QoE;
- Client Controller: receives the role to request video from the Streaming Unit set by the Fog Controller;
- Video Player: play video content on the user device;
- AAA Client: provides the authentication, authorization and accounting;
- QoE/QoS Meter: register QoS/QoE metrics on the user device;
- Energy Meter: register energy consumption of the video playback.

A specialized architecture for multimedia distribution was discussed in this chapter highlighting features and purposes each architecture module might work to improve user experience considering multimedia, the biggest responsible for the current and future data traffic application in the world. The deployment of this architecture extends the operational usage of the network core and end user devices to cope with a better user experience in a collaborative fashion considering gain and losses, especially for the mobile devices, which own a more constrained capacity.

# CHAPTER 5

## Experiments Methodology

This chapter describes the experimental scenarios and methodologies to evaluate the architecture. It uses real devices distributed in different geographical areas making experiments in a real network. Therefore, we experimentally evaluate the proposed architecture and explain more details in this chapter. In the end, we show the results of our tests. The experiment consider the AP or the mobile devices as a FN to provide cache service for mobile users. The experimental scenario composed of smartphones, an AP, and video service provider in three locations:

- (i) a smartphone connected to an AP for downloading VoD content from the video service provider at Tier 1;
- (ii) a smartphone connected to an AP for downloading VoD content cached in the AP at Tier 2;
- (iii) a smartphone downloading VoD content cached in another smartphone via WiFi-direct at Tier 3.

An on-demand video application running on the smartphone, i.e., ffplay [67], where users select and watch the video content when they choose. A buffer store a few seconds of the video content before start playing to minimize sporadic failures or delay fluctuations in the network transmission [69]. In highly congested wireless network with buffer overflow, and packet loss ratio, the video player may face with some re-buffering events, and extra time for initial buffering. This demands a performance evaluation of on-demand video transmission in such situation, but the current ffplay does not provide QoE measurements.

In this way, we used our developed Video Player to collect QoE metrics for on-demand video transmission, and thus make possible QoE evaluation. The client application used ffplay as base decoding software because most of Android video players uses it. In addition, it has support to play the videos coded in H.264. The client application requests multimedia content from the video service provider located at the Cloud, AP, or smartphone, those two last located in a fog.

The cloud computing provides a data center infrastructure with high computing resources permanently accessible by mobile users across the Internet. In the experiment, the cloud runs the video service provider to distribute the multimedia content for each client request. The video service provider streamed with ffserver [3], the video streamer software of ffmpeg [68]. The smartphone requests the video with different configurations of codecs and bitrates, cached on a fog node.

Thousands of users uploading/downloading multimedia content from Tier 1 will soon outstrip the bandwidth capacity and increase the users delay [11], since the network connection between mobile devices and cloud infrastructures worsen the QoE, especially in a dense wireless multimedia scenario. Emerging multimedia applications also requires cloud services at the network edge to meet the user needs. In this way, we consider fog nodes distributed in Tiers deployed closer to the mobile devices, where fog nodes can be deployed anywhere with a network connection. The AP act as a fog node to cache video closer to the user. In this way, the AP had started a ffserver, enabling the AP to stream the video for each client request.

The mobile devices become a fog node to relay the video content via D2D wireless communication, emulating crowned scenarios, such as in airports, railway stations, vehicular applications, and sport stadiums, where a group of mobile devices usually upload and download similar content in approximate time from the cloud using overloaded (and possible high cost) wireless links. Hence, a given mobile device becomes a fog node to download and share the content among its neighboring mobile devices using Wi-Fi direct.

## 5.1 Experiment 1 - Different Codec and Fully Congested Network

In this section, we analyzed the performance of the downloading on-demand video streaming configured with two codecs and three bitrates to evaluate their impact on the QoE and energy in distinct codec and fully congested network experiment. More specifically, we describe the scenarios, methodology, and metrics used to evaluate the quality level of transmitted video. We analyzed the impact of different codec and bitrate configuration on the video quality level downloaded from the cloud, fog, or smartphone.

### 5.1.1 Experiment 1 - Methodology

In the experiments, we consider the smartphone requesting the video from Tier 1, 2 and 3 in a scenario with and without network congestion (*i.e.*, congested and non-congested). More specifically, four smartphones request the video at given codec and bitrate configuration from Tier 1, 2 and 3 with and without network congestion. The smartphone in Tier 3 requests the video at given codec and bitrate configuration from another smartphone in a scenario without network congestion. We consider a video with duration of 75 seconds using two codecs (i.e., H.264 and H.265), at three bit rates (i.e., 400, 1000, and 2250 kbps), and with 30 frames per second. We repeated 30 times each experiment, and we the results provided a confidence interval of 95%.

We congested the networking using the iperf [70] running on a notebook to consume as much as possible the bandwidth from two different network points. In the experiments with a congested network, in the first 60 seconds we consider only the iperf running on a notebook, afterwards four smartphones requested the video at each 85 seconds for 30 times. The iperf request was set to a bandwidth target of 30 Mbps, packet size of standard 1500 bytes and one-way communication from the service provider to the end user. For each video transmission, we collected different QoE metrics and the energy consumption.

In terms of video quality evaluation, Quality of Service (QoS) schemes are not enough to infer the quality level of multimedia applications because they do not collect subjective human experience aspects of video content. For instance, we collected video playback start time as QoE metrics and RTT as QoS metric. In addition, we collected the battery level to evaluate how much energy the video demanded from the client devices decoding the two distinct codec H.265 and H.264.

Playback start time measures the buffering before a video starts to playout, i.e., typically includes the time taken to download the HTML page (or the manifest file), load the video player plugin, and to playback the initial part of the video. This metric has a significant influence on user experience and high values could result in the viewer abandoning the video completely. We reduced the number of background process, the display bright, and audio level. In this way, we ensure that most of the energy consumption occurred for video decoding process in the mobile device.

### 5.1.2 Experiment 1 - Scenario

In summary, we set the Tier 1 at Federal University of Rio Grande do Sul using the FIBRE testbed depicted by Figure 9 . In the first part of the experiment, we set a Streaming Unit at Tier 1 running the Fog Application and requested video contents from the Video Service Host and collected QoE and QoS metrics with four smartphone running the video player developed by us. Thus, our QoE/QoS and energy meters were running in the mobile device while the video played.

In second part of the experiment, we set Tier 2 at Federal University of Pará as the AP fog node running the Media Server Application. The fog node hosted the same video contents at the Cache Unit and requested them from the Streaming Unit to the smartphones. The mobile devices executed our video player as well in order to

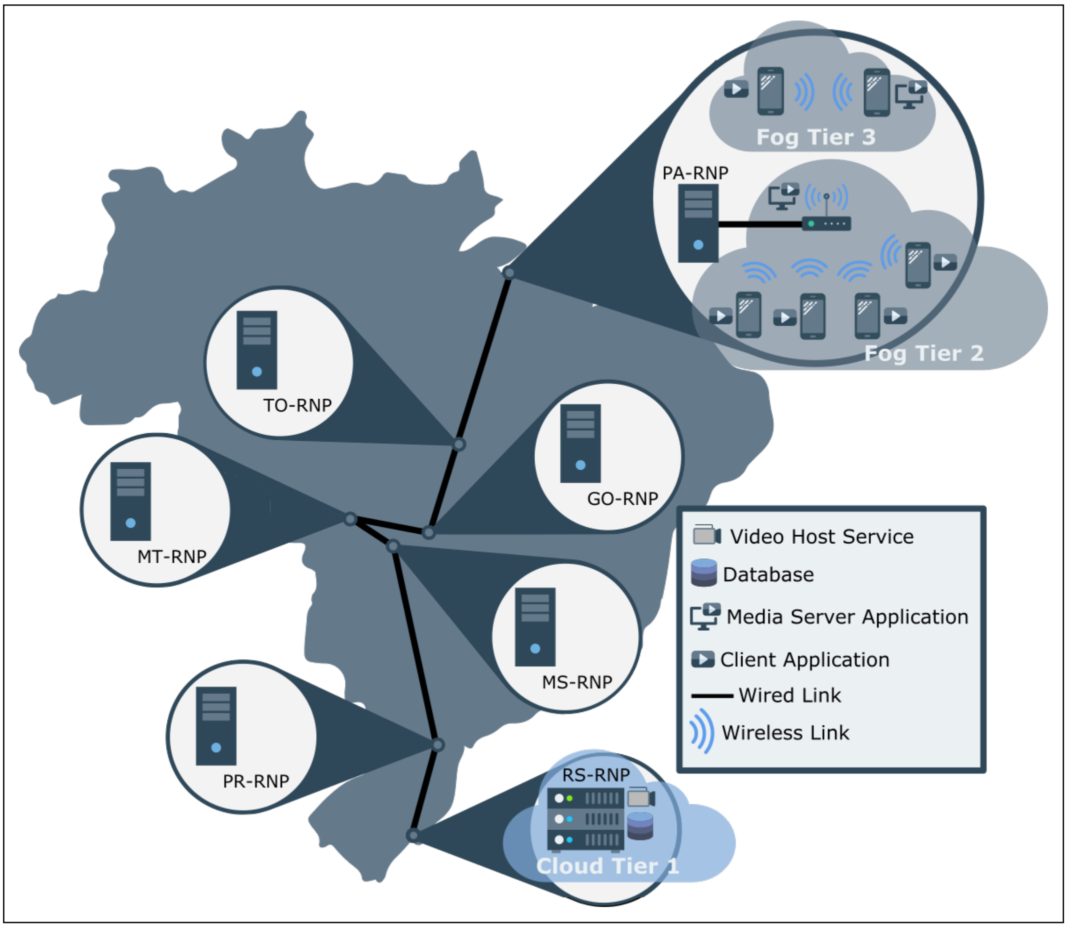


Figure 9: Distinct Codec and Fully Congested Network Scenario

collect the same metrics. In this case, the video contents did not come through the fiber optical network of FIBRE testbed[1] preventing delays and bandwidth constraints of this infrastructure.

In the last part of the experiment, we set Tier 3 as a D2D communication between two smartphones. One of the smartphones executed the Media Server Application, thus running the Relay Connection Manager, Cache Unit and Streaming Unit elements in order to distribute video contents to the other smartphone. The other mobile device executed our video player as well to collect again the same metrics but with a dedicated link without any concurrent traffic.

### 5.1.3 Experiment 1 - Results

Figure 10 shows the playback start time for downloading the video with different bitrates from Tier 1, 2, and 3 in a scenario with and without network congestion. In case of network congested, the video player starts to play the video downloaded from Tier 1 after 18 seconds for video configured with bitrate of 400 kbps, and increases to 45 seconds for video configured with bitrate of 2250 kbps. On the other hand, downloading the video

[1] www.fibre.org.br

from Tier 2 and 3 with different bitrates has the playback start time stable and about 1 second. This behavior happens because of the Internet connection between AP and the cloud largely influence the network performance.

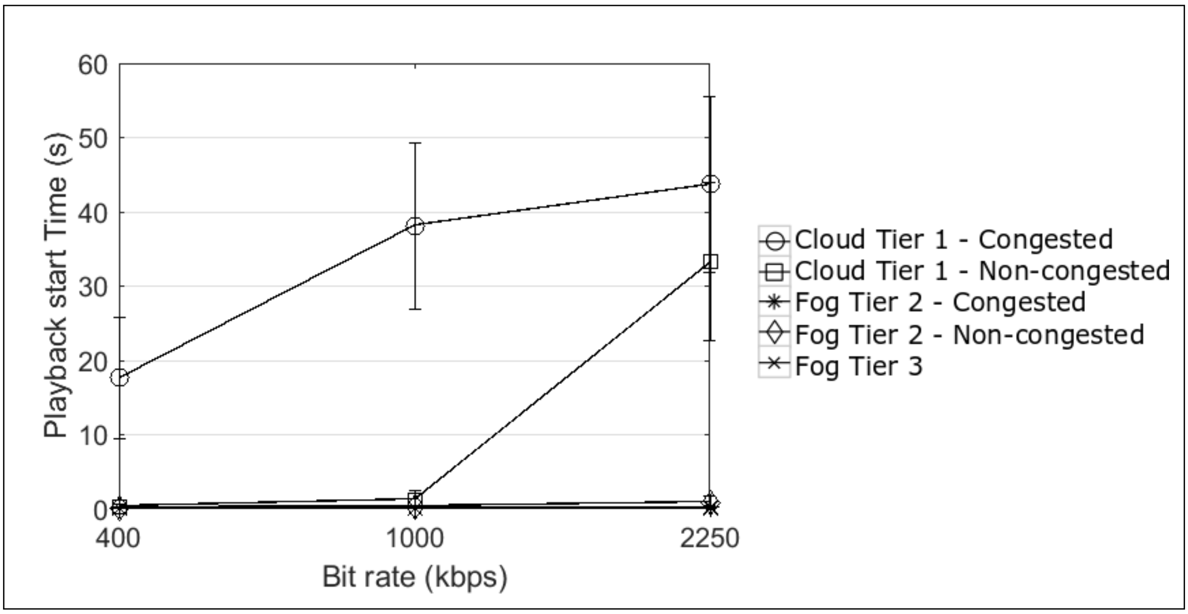


Figure 10: Playback start time for downloading the video with different bitrates from the cloud, fog, and another smartphone in a scenario with and without network congestion

Figure 11 shows RTT for smartphones downloading the video with different bitrates from the three Tiers in a scenario with and without network congestion. We observe that Tier 1 has a slightly worse RTT performance in case of downloading the video in a congested situation for the video configured with bitrates of 400 and 1000 kbps. On the other hand, we observe higher RTT value and variation for the bitrate of 2250 kbps, since it demands more data transmission compared to the bitrates of 400 and 1000 kbps.

For video downloaded from the Tier 2 without network congestion, we observe that the RTT increases as long as the bitrate increases, since higher bitrate demands the more data transmission compared to lower bitrates. In addition, downloading the video from the fog presented an unstable RTT, because of the lower RTT for 1000 and 2250 kbps. This behavior happened because the AP has the primary function to provide wireless network access, and we started a streaming unit as a secondary service on this hardware, and this might cause an unstable RTT performance. Finally, downloading the video cached in a given smartphone on Tier 3 has similar and constant RTT performance regardless the bitrates. This behavior occurs because of the absence of concurrent network traffic.

As it was expected, we conclude that downloading the video from Tier 1 has the worst performance compared to Tiers 2 and 3 regarding videos in different bitrates. For instance, downloading the content from Tier 1 in a congested network has a RTT value 2 times higher compared to both Tier 2 and 3. This performance confirms the importance to bring content closer to meet user needs for on-demand video distribution with adequate QoE.

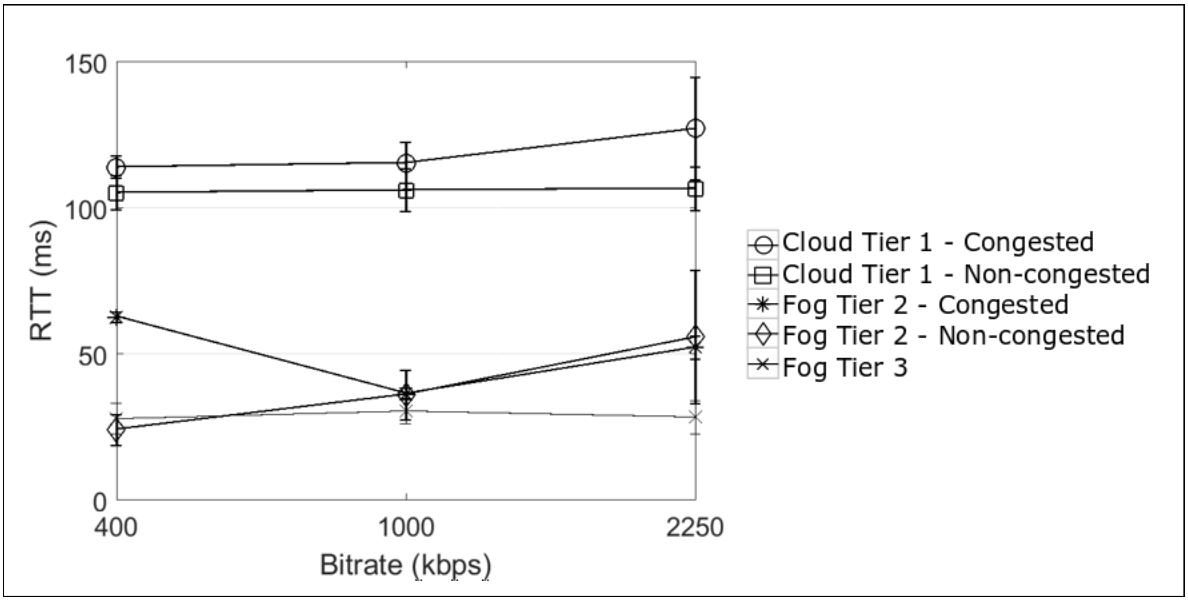


Figure 11: RTT for downloading the video with different bitrates from the cloud, fog, and another smartphone in a scenario with and without network congestion

Figure 12 shows the energy consumption demanded from the client devices to receive and play the video coded into H.264 and H.265 at bitrate values of 400, 1000 and 2250 kbps. Based on our energy results, we conclude that H.264 demands more energy for decoding as long as the bit rate increases. More specifically, H.265 demands almost the same energy for the bitrate of 400 kbps coded at H.264. On the other hand, video coded at H.265 at the bitrate of 2250 kbps demands more energy consumption compared to the video coded at H.264, since the client device needs to perform more complex task for decoding. In this way, users must be aware of this different energy consumption between different codecs and bitrates. It is important to mention that energy assessment could be better, but our measurements enable us to infer the global energy consumption for videos coded at H.264 and H.265 with different bitrates at the client side.

## 5.2 Experiment 2 - Variable Network Congestion Experiment

In this section, we describe the experiment 2 scenarios, methodology, and metrics used to evaluate the quality level of transmitted videos considering the proposed architecture in a variable network congestion experiment. Then, we analyze the impact of different network congestion level on the video quality downloaded from different tiers of the proposed architecture

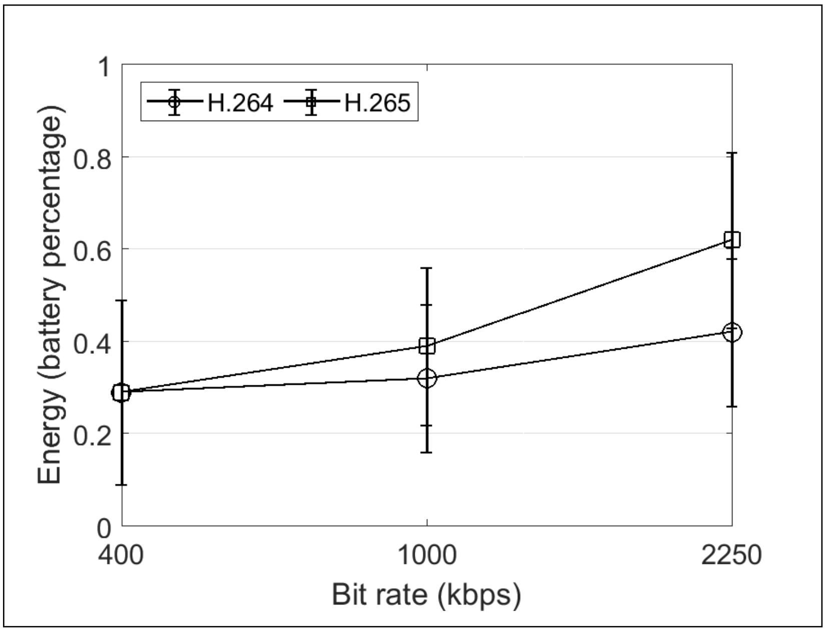


Figure 12: Energy consumption video coded into H.264 and H.265 at bitrate values of 400, 1000 and 2250 kbps

### 5.2.1 Experiment 2 - Methodology

This experiment consider an HTTP server application to enable the mobile device to stream the video content for each request from the neighbor mobile device. Thus, the video content transmission occurs using WiFi-direct is composed of a general-purpose hardware to run the streaming unit with higher or lower computing power. On the other hand, Tier 3, i.e., Icarus WiFi Node in our experiments, has the primary function to provide wireless network access, and we started a streaming unit as a secondary service on this hardware.

### 5.2.2 Experiment 2 - Scenario

In summary, we set the Tier 1 at Federal University of Minas Gerais using the FIBRE testbed depicted by Figure 13. In the first part of the experiment, we set a Streaming Unit at Tier 1 running the Fog Application and requested video contents from the Video Service Host and collected QoE and QoS metrics with a unique smartphone running the video player developed by us. Thus, our QoE/QoS and energy meters were running in the mobile device while the video played.

In second part of the experiment, we set Tier 2 at Federal University of Rio Grande do Sul as the AP fog node running the Media Server Application. The fog node hosted the same video contents at the Cache Unit and requested them from the Streaming Unit to the smartphone. The mobile device executed our video player as well in order to collect the same metrics. In this case, the video contents did not come through the fiber optical network of FIBRE testbed preventing delays and bandwidth constraints of this

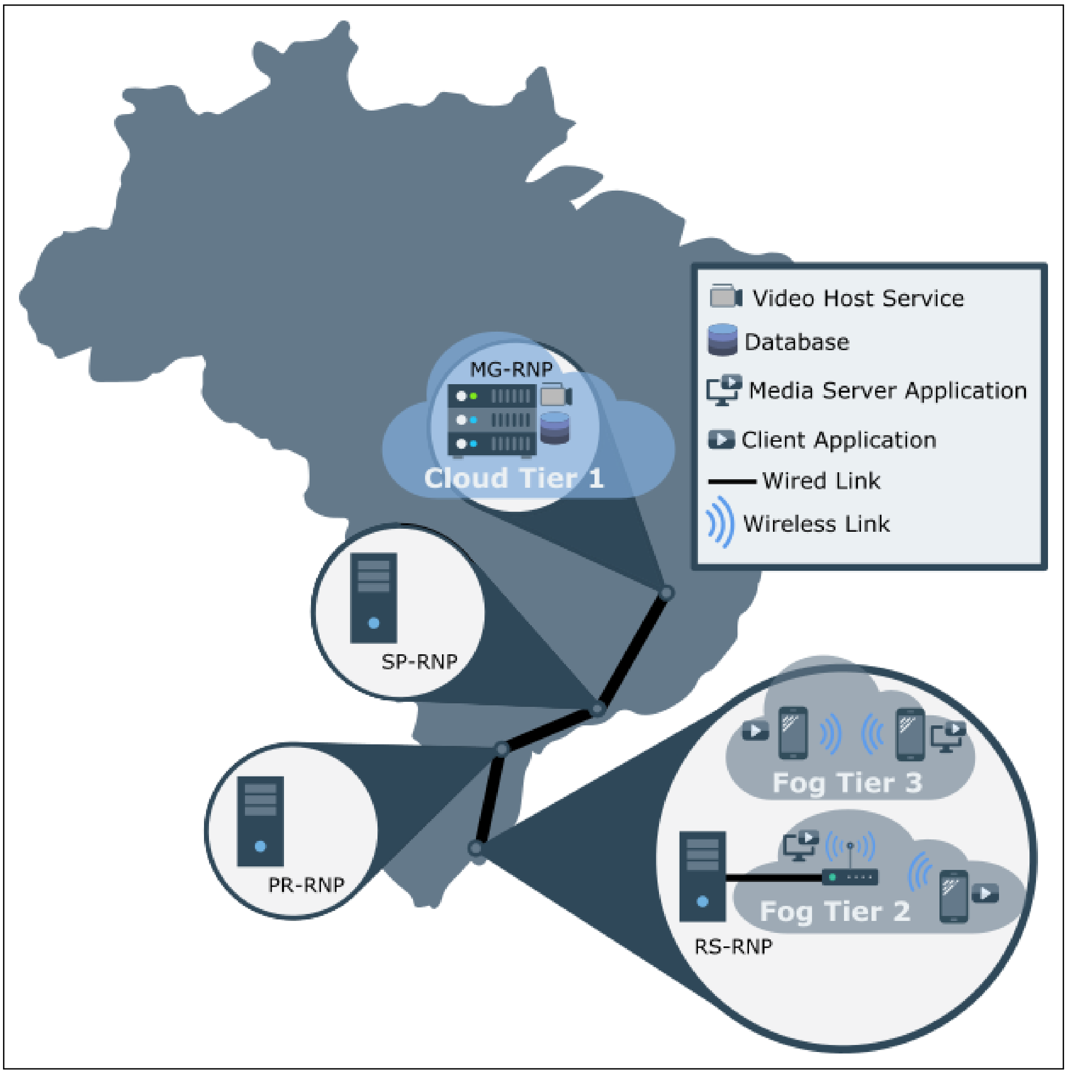


Figure 13: Variable Network Congestion Experiment Scenario

infrastructure.

In the last part of the experiment, we set Tier 3 as a D2D communication between two smartphones. One of the smartphones executed the Media Server Application, thus running the Relay Connection Manager, Cache Unit and Streaming Unit elements in order to distribute video contents to the other smartphone. The other mobile device executed our video player as well to collect again the same metrics but with a dedicated link without any concurrent traffic.

### 5.2.3 Experiment 2 - Results

We analyzed RTT, playback start time, and duration of freezes in a scenario where the Video Player downloads the video content from the Tiers 1, 2, and 3 with different background traffic (i.e., 0, 1, 2, 3, 4, and 5 Mbit/s). For this experiment, we encoded a video with rate of 30 frames per seconds with the time length of 75 seconds. In addition, we encoded the video with codecs H.264 and H.265 into six different configurations: two codecs and at the bit rates of 400, 1000, and 2250 kbps. We repeated 30 times each experiment, and we the results provided with a confidence interval of 95%. Figures 14, 15 and 16 depict the results of playback start time, and duration of freezes, and RTT, in the first experiment, respectively.

We observe a lower and constant playback start time for downloading the video from Tier 2 and 3 compared to other tiers by analyzing the Figure 14. This means that the client application starts to play the video for the user before 600 ms regardless the background traffic. The gain, regarding playback start time, for Tier 2 and 3 increases in a dense and congested scenario compared to Tier 1. For instance, the playback start time for Tier 3 is 5%, and 65% better than for Tiers 2, and 1, respectively, for the background traffic of 5 Mbit/s. This behavior occurs because the dedicated D2D connection between two smartphones without concurrent traffic. Tier 1 has longer playback start time compared to other tiers, and this value worsens in case of higher background traffic. This behavior happens due to a higher typical delay in Tier 1 with longer variation compared to other tiers.

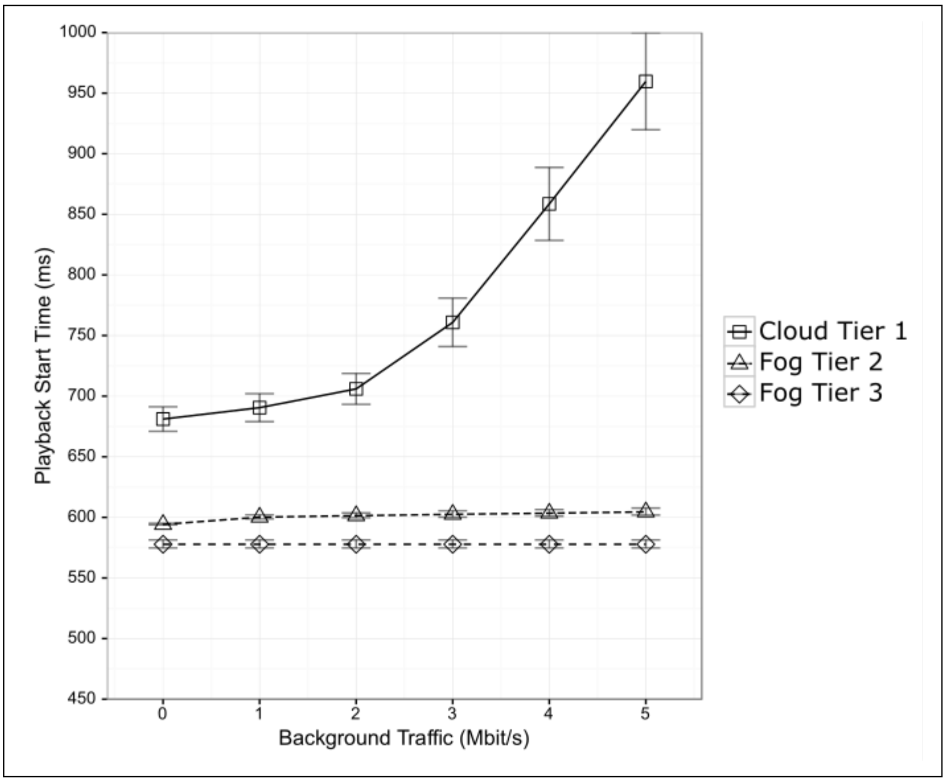


Figure 14: Evaluation results for the playback start time downloading the video from Tiers 1, 2, and 3 with different background traffic

Tier 2 has a constant playback start time, while Tier 1 increases the value as long as the background increases. This performance event happened because of the longer path to send data depending of more actors using the network. Tier 2 locates in local network of a private or mobile network operator where the operator does not usually share its network links, thus a better QoS. Tier 2 network elements range from datacenters to radio access devices but in our evaluation we hosted video content only at the radio access devices.

Figure 15 depicts the freeze times for the first experiment. A freeze occurs when the buffer gets depleted and the player waits for a partial re-buffering before resuming the playback, i.e., the playback of the video temporarily stalls to receive a minimum amount playable video data. In this sense, all tiers have freeze times around 2 second for the background traffic of 0 Mbit/s, but Tiers 1 and 2 struggle more for re-buffering events proportionally as the background traffic increases. In our experiments, Tier 3 enables to play the video with freeze times of around 2 second, while the Tier 1 has freeze times higher than 23 seconds for the background traffic higher than 1 Mbit/s. These re-buffering events take at least 11 seconds for resuming the video playback for Tiers 2. More specifically, Tier 1 freezes more time during the video transmission compared to Tier 2, despite Tier 3, which enabled the user to watch the video quicker compared to Tier 2 independently of the background traffic. Tier 2 performance stays between Tiers 1 and 3 because the background traffic does not affect the relay network since the smartphone creates a new network only for relaying.

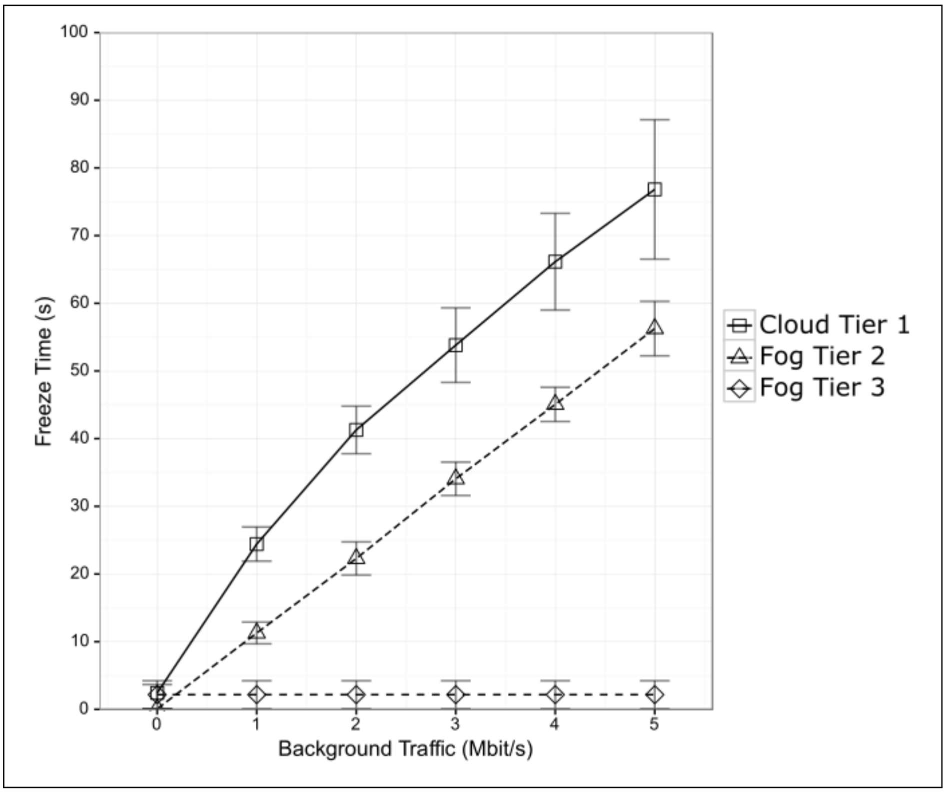


Figure 15: Evaluation results for the freeze time downloading the video from Tiers 1, 2, and 3 with vifferent background traffic

Figure 16 shows the RTT measured in the first experiment. The smartphone downloading the video from the Tier 1 has the worst RTT performance, especially in a dense and congested HetNets scenario. RTT measured for Tier 1 with background traffic of 5 Mbit/s is 50% higher than for Tier 1 with 0 Mbit/s. This behavior occurs because the connection between the smartphone and the Tier 1 largely affects the network performance, where the typical delay measured in our experimental scenario between smartphone and Tier 1 is about 7 times higher than between smartphone and Tier 2. Finally, downloading the video from Tier 3 has a constant RTT performance, since the background traffic does not impact the D2D wireless connection.

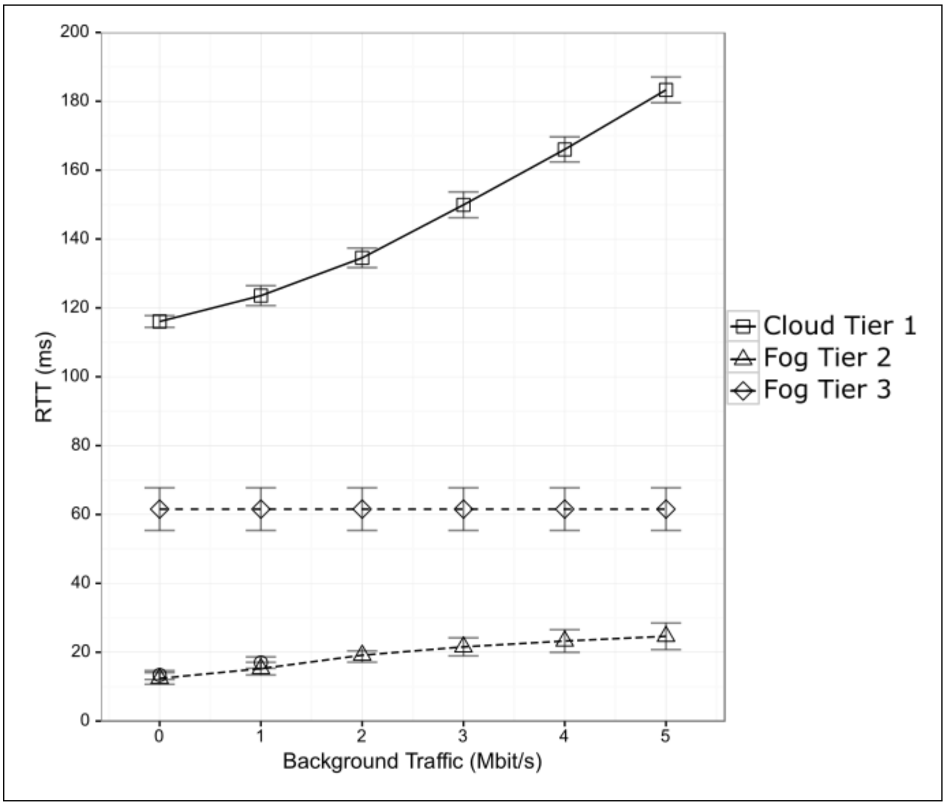


Figure 16: Evaluation results for the RTT downloading the video from Tiers 1, 2, and 3 with different background traffic

We also measured the energy consumption of a smartphone relaying the video via WiFi-direct at Tier 3. More specifically, we consider a smartphone with a Quad-Core 2.5GHz, RAM of 2GB, and storage of 16GB as a relay node. This experiment consider the smartphone in three settings: standby mode, and running the HTTP server to relay the video for one or two clients. We collected the energy consumption every 8 seconds during 1 hour via an oscilloscope Tektronix MDO3012. We sampled with a granularity of 1000 discrete events at each second and the curve resembles a normal distribution. In Table 4, we present the mean, standard deviation, minimum, and maximum with respect to each setting. We note that the energy consumption to relay the video content almost doubles the energy consumption in the standby mode considering one device. This happens because the smartphone enables the WiFi network interface and the HTTP server for the video streaming. Moreover, the measurements showed no significant difference between relaying for one or two clients.

The energy consumption in a smartphone affects the battery life becoming especially important. In this sense, as previously stated, relaying increases the energy consumption. The second experiment, we asset that the relay mode can consume at most 25% of a 2800 mAh battery during 1 hour. Therefore, for videos up to 2 minutes, the relay consumption waste less than 1% of a battery with such capacity. It seems adequate

Table 4: Energy Consumption for Cache Schemes in Different Levels

| **Setting** | **Average (W)** | **Minimum (W)** | **Maximum (W)** | **Standard Deviation** |
|---|---|---|---|---|
| Standby | 0.7542214 | 0.6112 | 2.9280 | 0.2838545 |
| One Device | 1.4491829 | 0.6112 | 6.9472 | 0.8618381 |
| Two Devices | 1.4774190 | 0.6112 | 6.4896 | 0.8717943 |

for a given user cooperate with the neighbors or incentive energy spending in change for a reward.

## 5.3 Chapter Conclusions

This chapter described the experimental scenario, including the route over the Internet to distribute the devices in the cloud and end user considering a reasonable geographical distance between them. The applications executed in every Tiers and provided trustworthy results for the architecture evaluation.

Video delivery over mobile wireless networks will take a significant percentage of overall global data traffic. In this sense, our multi-tier architecture can provide significant benefits over traditional cloud environments in the context of mobile networking. In this context, we compared the performance of downloading the video coded into H.264 and H.265 with different bitrates. We also compared the performance of downloading such videos from Tier 1, 2, and 3 in terms of RTT, QoE, and energy.

The experimental results confirm that downloading the video cached in Tier 2 and 3 brings benefits for the client. As long as the bitrate increases, the network or user experience performance tends to increase, since higher bit rate requires more data to transmit, but it provides more QoE. Regarding to the video coded, we observed that for higher bitrates H.265 offers twice encode efficiency than H.264, but this comes with higher decoding energy consumption. These results bring important understanding of the use of cache and transcoding schemes in wireless multimedia networking scenarios.

# CHAPTER 6

## Conclusion

Users and their mobile devices have brought a great challenge to researchers and companies service providers in recent years. In this sense, Fog Computing establishes an infrastructure that can solve many of these challenges by offering the infrastructure capable of withstanding the high demand for processing and traffic generated for different types of applications.

This dissertation has proposed as an architecture that integrates fog computing to support video on demand streaming with the objective of improve QoE considering network degradation and multiple fog deployment areas. From this initial proposal, we developed study tools to evaluate the gains of the architecture and measure which requirements can meet in the current network infrastructure.

The studies provided a conceptual basis for the formulation of the architectures proposed here and for the creation of QoE/QoS and energy tools for Android operating system. Furthermore, we tested part of the architecture in experimental environment assessing real world data. In the experiment 1, the playback start time could be reduced almost 18 seconds for 400 kbps and 38 seconds for 1000 kbps when request content a congested link from cloud tier 1 and fog tier 2. In addition, the delay could be reduced to almost half comparing to the delay from both experiments in the cloud.

For experiment 2, the variable network congestion showed how cloud tier 1 can be degraded when the concurrent traffic increases in terms of playback start time. However, fog tier 2 and 3 had a stable behavior. Regarding freeze times, the time increasing was almost linear with the background traffic rising scale, but this result showed poor QoE performance on cloud tier 1 and fog tier 2. Otherwise, for fog tier 3, the QoE performed regular and acceptable.

The first tests confronted a scenario in which the mobile device shared a link with background traffic in order to create concurrency in the communication link. The results obtained from experiments show how much the edge and the D2D communication can the enhance QoE and QoS at a low cost for the smartphones to collaboratively help the data dissemination in the network. Tier 2 and 3 surpass the link degradation when congested.

The performance of the second stage we designed a slightly distinct scenario from the first experiment. We evaluated almost the same metrics for QoE and QoS, however the energy consumption for H.265 decoding on mobile device highlights and shows how it can contribute to reduce the network traffic with similar video quality and energy consumption. Many other aspects and challenges of this architecture must still be implemented as a goal to have a scalable infrastructure that supports robust and effective the other elements of the architecture, thus leading to an automated modules of the architecture for video on demand streaming, such as the cache, transcoding and device fog deployment mechanisms.

## 6.1 Challenges

Deploy the architecture in an experimental environment brought many challenges in order to perform the tests. The overall results had a considerable difference when collected in different day hours and we tried most regular as possible in any repetitions of the experiments. Moreover, we have experienced a fluctuation of the wireless network throughput capacity with no apparent reasons but presented regular operation most of the time. It happened for the conventional wireless AP produced by TP-Link and also for Icarus WiFi Node.

The wireless channel presented as a non trust able environment since the presence of many interferences despite our efforts for a controlled environment. We also faced difference when the smartphone performed as relay node. In the first experiment, we had a RTT of 30 seconds which was very similar with the conventional AP when idle. However, the smartphone in the seconds experiment present a constant RTT of nearly 60 ms, almost twice of the first experiment. We account this as a wireless card behavior and the operational system processes concurrency.

Finally, the facilities to deploy the datacenter was challenging to arrange and configure. The remote configuration of the devices

## 6.2 Academical Production

Part of the results of this work were published on:

- **H. SANTOS**; D. ROSARIO; E. CERQUEIRA; J. CAMARGO; M. SCHIMUNECK; J. NOBRE; C. BOTH. "A Comparative Analysis of H.264 and H.265 with

Different Bitrates for on Demand Video Streaming". *9th Latin America Networking Conference 2016*. LANC - 2016.

And part of the results of this work were submitted on:

- **H. SANTOS**, D. ROSARIO, J. NOBRE, C. BOTH; M. NOGUEIRA and E. CERQUEIRA "An Integrated Fog Architecture for Advanced Multimedia Distribution and Future Direction". *IEEE Communication Magazine* - 2018.

# References

[1] M. M. Lopes, "Arquitetura e mecanismos para migração de máquinas virtuais na computação em névoa," Ph.D. dissertation, State University of Campinas, 2017.

[2] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of quality of experience of video-on-demand services: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 401–418, 2016.

[3] FFMPEG, "ffserver documentation," https://ffmpeg.org/ffserver.html, 2016, access in: Jan 2018.

[4] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "Qoe-driven cache management for http adaptive bit rate streaming over wireless networks," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1431–1445, 2013.

[5] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, 2016.

[6] Ericsson, "Mobile traffic analysis by application." Ericsson, White Paper, Tech. Rep., 2017.

[7] 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*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16. [Online]. Available: http://doi.acm.org/10.1145/2342509.2342513

[8] Y. C. H. et al., "Mobile edge computing a key technology towards 5g." ETSI, Tech. Rep., 2015.

[9] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct 2009.

[10] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, December 2014.

[11] M. A. Marotta, L. R. Faganello, M. A. K. Schimuneck, L. Z. Granville, J. Rochol, and C. B. Both, "Managing mobile cloud computing considering objective and subjective perspectives," *Computer Networks*, vol. 93, pp. 531–542, 2015.

[12] M. Jo, T. Maksymyuk, B. Strykhalyuk, and C.-H. Cho, "Device-to-device-based heterogeneous radio access network architecture for mobile cloud computing," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 50–58, 2015.

[13] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, "Fog computing in healthcare–a review and discussion," *IEEE Access*, vol. 5, pp. 9206–9222, 2017.

[14] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled iot networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 14–20, 2017.

[15] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, 2015, pp. 37–42.

[16] Ericsson, "Ericsson mobile report: On the pulse of the networked society." Ericsson, White Paper, Tech. Rep., 2013.

[17] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[18] H. Santos, D. Rosário, E. Cerqueira, J. Camargo, M. Schimuneck, J. Nobre, and C. Both, "A comparative analysis of h. 264 and h. 265 with different bitrates for on demand video streaming," in *Proceedings of the 9th Latin America Networking Conference*. ACM, 2016, pp. 53–58.

[19] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020." Cisco, White Paper, Tech. Rep., 2017.

[20] S. Andreev, A. Pyattaev, K. Johnsson, O. Galinina, and Y. Koucheryavy, "Cellular traffic offloading onto network-assisted device-to-device connections," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 20–31, 2014.

[21] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[22] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, 2010.

[23] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, "On qos-aware scheduling of data stream applications over fog computing infrastructures," in *Computers and Communication (ISCC), 2015 IEEE Symposium on*. IEEE, 2015, pp. 271–276.

[24] M. Sheng, W. Han, C. Huang, J. Li, and S. Cui, "Video delivery in heterogenous crans: architectures and strategies," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 14–21, 2015.

[25] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.

[26] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 1–14.

[27] F. Song, Z.-Y. Ai, J.-J. Li, G. Pau, M. Collotta, I. You, and H.-K. Zhang, "Smart collaborative caching for information-centric iot in fog computing," *Sensors*, vol. 17, no. 11, p. 2512, 2017.

[28] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[29] FFMPEG, "Ffmpeg and h.265 encoding guide," https://trac.ffmpeg.org/wiki/Encode/H.265, 2016, access in: Jan 2018.

[30] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2831347.2831354

[31] B. I. Ismail, E. M. Goortani, M. B. A. Karim, W. M. Tat, S. Setapa, J. Y. Luke, and O. H. Hoe, "Evaluation of docker as edge computing platform," in *2015 IEEE Conference on Open Systems (ICOS)*, Aug 2015, pp. 130–135.

[32] B. Juurlink, M. Alvarez-Mesa, C. Chi, A. Azevedo, C. Meenderinck, and A. Ramirez, "Understanding the application: An overview of the h.264 standard," in *Scalable Parallel Programming Applied to H.264/AVC Decoding*, ser. SpringerBriefs in Computer Science. Springer New York, 2012, pp. 5–15. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-2230-3_2

[33] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, July 2003.

[34] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, Jan 2005.

[35] INTERNATIONAL TELECOMMUNICATION UNION, "Definitions of terms related to quality of service," RECOMMENDATION ITU-T E.800, 30 p, Geneva, 2008.

[36] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A framework for qos-based routing in the internet," United States, 1998.

[37] H.-J. Zepernick and U. Engelke, "Quality of experience of multimedia services: past, present, and future," in *Adjunct Proceedings of the 9th European Interactive TV Conference (EuroITV 2011)*, 2011, pp. 115–119.

[38] S. Möller and A. Raake, *Quality of experience: advanced concepts, applications and methods.* Springer, 2014.

[39] R. J. Sternberg, *Psicologia Cognitiva*, 5th ed. São Paulo: Cengage Learning, 2010.

[40] H. J. Kim and S. G. Choi, "A study on a qos/qoe correlation model for qoe evaluation on iptv service," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 2. IEEE, 2010, pp. 1377–1382.

[41] M. Saxena, U. Sharan, and S. Fahmy, "Analyzing video services in web 2.0: a global perspective," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video.* ACM, 2008, pp. 39–44.

[42] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 2001–2014, 2013.

[43] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on.* IEEE, 2012, pp. 1–6.

[44] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in *Multimedia (ISM), 2011 IEEE International Symposium on.* IEEE, 2011, pp. 494–499.

[45] Y. Qi and M. Dai, "The effect of frame freezing and frame skipping on video quality," in *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06. International Conference on.* IEEE, 2006, pp. 423–426.

[46] B. Lewcio, B. Belmudez, T. Enghardt, and S. Möller, "On the way to high-quality video calls in future mobile networks," in *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on.* IEEE, 2011, pp. 43–48.

[47] M. Montenovo, A. Perot, M. Carli, P. Cicchetti, and A. Neri, "Objective quality evaluation of video services," in *Second International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006.

[48] S. Winkler, A. Sharma, and D. McNally, "Perceptual video quality and blockiness metrics for multimedia streaming applications," in *Proceedings of the international symposium on wireless personal multimedia communications*, 2001, pp. 547–552.

[49] W. Lin and C.-C. J. Kuo, "Perceptual visual quality metrics: A survey," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 297–312, 2011.

[50] U. Engelke and H.-J. Zepernick, "Perceptual-based quality metrics for image and video services: A survey," in *Next Generation Internet Networks, 3rd EuroNGI Conference on.* IEEE, 2007, pp. 190–197.

[51] J. You, U. Reiter, M. M. Hannuksela, M. Gabbouj, and A. Perkis, "Perceptual-based quality assessment for audio–visual services: A survey," *Signal Processing: Image Communication*, vol. 25, no. 7, pp. 482–501, 2010.

[52] L. Yitong, S. Yun, M. Yinian, L. Jing, L. Qi, and Y. Dacheng, "A study on quality of experience for adaptive streaming service," in *Communications Workshops (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 682–686.

[53] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 362–373.

[54] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, "Ehopes: Data-centered fog platform for smart living," in *2015 International Telecommunication Networks and Applications Conference (ITNAC)*, Nov 2015, pp. 308–313.

[55] D. Nguyen, Z. Shen, J. Jin, and A. Tagami, "Icn-fog: An information-centric fog-to-fog architecture for data communications," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[56] W. Zhang, Z. Zhang, and H. C. Chao, "Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 60–67, DECEMBER 2017.

[57] R. Vilalta, V. Lopez, A. Giorgetti, S., V. Orsini, L. Velasco, R. Serral-Gracia, D. Morris, S. D. Fina, F. Cugini, P. Castoldi, A. Mayoral, R. Casellas, R. Martinez, C. Verikoukis, and R. Munoz, "Telcofog: A unified flexible fog and cloud computing architecture for 5g networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 36–43, 2017.

[58] H. Ahlehagh and S. Dey, "Video caching in radio access network: Impact on delay and capacity," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. IEEE, 2012, pp. 2276–2281.

[59] A. Gomes, T. Braun, and E. Monteiro, "Enhanced Caching Strategies at the Edge of LTE Mobile Networks," in *Proceedings of the IFIP Networking Conference and Workshops (IFIP Networking)*, May 2016, pp. 341–349.

[60] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "An energy and delay-efficient partial offloading technique for fog computing architectures," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[61] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "Fogroute: Dtn-based data dissemination model in fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 225–235, Feb 2017.

[62] Y. Wang and X. Lin, "User-provided Networking for QoE Provisioning in Mobile Networks," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 26–33, August 2015.

[63] D. Karvounas, A. Georgakopoulos, K. Tsagkaris, V. Stavroulaki, and P. Demestichas, "Smart Management of D2D Constructs: an Experiment-based Approach," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 82–89, April 2014.

[64] X. Wang, M. Chen, Z. Han, D. O. Wu, and T. T. Kwon, "Toss: Traffic offloading by social network service-based opportunistic sharing in mobile social networks," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 2346–2354.

[65] H. A. Pedersen and S. Dey, "Enhancing mobile video capacity and quality using rate adaptation, ran caching and processing," *IEEE/ACM Transactions on Networking (TON)*, vol. 24, no. 2, pp. 996–1010, 2016.

[66] B. O. S. T. Force, "ijkplayer," https://github.com/Bilibili/ijkplayer, 2016.

[67] FFMPEG, "ffplay documentation," https://ffmpeg.org/ffplay.html, 2016, access in: Jan 2018.

[68] ——, "A complete, cross-platform solution to record, convert and stream audio and video," https://ffmpeg.org, 2016, access in: Jan 2018.

[69] O. B. Maia, H. C. Yehia, and L. de Errico, "A concise review of the quality of experience assessment for video streaming," *Computer Communications*, vol. 57, pp. 1 – 12, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366414003648

[70] J. Dugan, S. Elliott, B. A. Mah, J. Poskanzer, and K. Prabhu, "iperf - the ultimate speed test tool for tcp, udp and sctp," https://iperf.fr/, 2016, access in: Jan 2018.