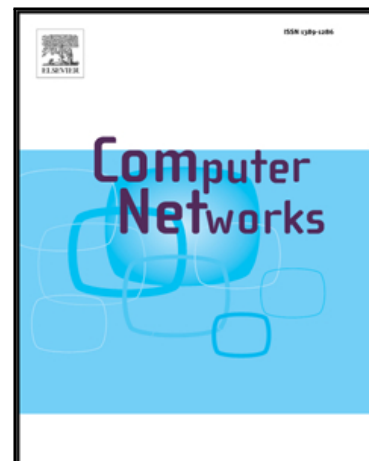


Journal Pre-proof

A Multi-Tier Fog Content Orchestrator Mechanism with Quality of Experience Support

Hugo Santos, Derian Alencar, Rodolfo Meneguette, Denis Rosário, Jéferson Nobre, Cristiano Both, Eduardo Cerqueira, Torsten Braun

PII: S1389-1286(20)30444-8
DOI: <https://doi.org/10.1016/j.comnet.2020.107288>
Reference: COMPNW 107288



To appear in: *Computer Networks*

Received date: 23 March 2020
Revised date: 14 April 2020
Accepted date: 30 April 2020

Please cite this article as: Hugo Santos, Derian Alencar, Rodolfo Meneguette, Denis Rosário, Jéferson Nobre, Cristiano Both, Eduardo Cerqueira, Torsten Braun, A Multi-Tier Fog Content Orchestrator Mechanism with Quality of Experience Support, *Computer Networks* (2020), doi: <https://doi.org/10.1016/j.comnet.2020.107288>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier B.V.

A Multi-Tier Fog Content Orchestrator Mechanism with Quality of Experience Support

Hugo Santos^{a,b,*}, Derian Alencar^a, Rodolfo Meneguette^c, Denis Rosário^a, Jéferson Nobre^d, Cristiano Both^e, Eduardo Cerqueira^a, Torsten Braun^b

^a*Federal University of Pará, Brazil*

^b*University of Bern, Switzerland*

^c*University of São Paulo, Brazil*

^d*Federal University of Rio Grande do Sul, Brazil*

^e*University of Vale do Rio dos Sinos, Brazil*

Abstract

Video-on-Demand (VoD) services create a demand for content orchestrator mechanisms to support Quality of Experience (QoE). Fog computing brings benefits for enhancing the QoE for VoD services by caching the content closer to the user in a multi-tier fog architecture, considering their available resources to improve QoE. In this context, it is mandatory to consider network, fog node, and user metrics to choose an appropriate fog node to distribute videos with QoE support properly. In this article, we introduce a content orchestrator mechanism, called of Fog4Video, which chooses an appropriate fog node to download video content. The mechanism considers the available bandwidth, delay, and cost, besides the QoE metrics for VoD, namely number of stalls and stalls duration, to deploy VoD services in the opportune fog node. Decision-making acknowledges periodical reports of QoE from the clients to assess the video streaming from each fog node. These values serve as inputs for a real-time Analytic Hierarchy Process method to compute the influence factor for each parameter and compute the QoE improvement potential of the fog node. Fog4Video is executed in fog nodes organized in multiple tiers, having different characteristics to provide VoD services. Simulation results demonstrate that Fog4Video transmits adapted

*Corresponding author

Email addresses: hugosantos@ufpa.br (Hugo Santos), derian.alencar@itec.ufpa.br (Derian Alencar), meneguette@icmc.usp.br (Rodolfo Meneguette), denis@ufpa.br (Denis Rosário), jcnobre@inf.ufrgs.br (Jéferson Nobre), cbboth@unisinos.br (Cristiano Both), cerqueira@ufpa.br (Eduardo Cerqueira), torsten.braun@inf.unibe.ch (Torsten Braun)

videos with 30% higher QoE and reduced monetary cost up to 24% than other content request mechanisms.

Keywords: Content Orchestrator, Multi-Tier Fog Computing, Quality of Experience

1. Introduction

Multimedia service providers will disrupt the TV and video industry, where big multimedia players, such as YouTube, Amazon, and Netflix, will soon replace the current broadcasters [1]. The market players are attracting thousands of new customers every day, where they are expecting to access video-on-demand (VoD) services with high Quality of Experience (QoE) while optimizing their needs in terms of image definition and playback smoothness [2]. Therefore, VoD providers require new mechanisms to deliver high-quality video streaming in dynamic and adaptive cellular/wireless networks during usage peaks, for instance, by using HTTP streaming [3]. This behavior is especially true during the staying home over the COVID-19 concerns, where several European Internet Exchange Point (IXPs) reached an all-time peak [4]. Due to the lack of these mechanisms in 4G and 5G cellular networks, many providers end up lowering the video quality to reduce network traffic for delivery content with smoother playback. However, users end up abandoning the service due to stalls, rebufferings, playback average, and bitrate variability [2, 5].

In the last few years, caching services and fog computing have gained a lot of attention in both industry and academia as new schemes to improve the VoD delivery to mobile users [6]. Fog computing extends cloud computing services with (ultra) low latency, and additional computing, storage, and networking resources closer to the user. Fog computing offers VoD services cached closer to the user, providing the transmission requirements properly in terms of delay and bandwidth [7]. In this sense, a multi-tier fog computing environment enables to execute VoD services at each level to match the topology and distributed workload properties of VoD applications, while meeting QoE requirements [6]. However, how, where, and when to orchestrate the usage of fog nodes for VoD services in a multi-tier fog computing scenario is not a trivial task and it is still an open issue [8].

The cloud and fog nodes work collaboratively, where cloud computing provides VoD services and also orchestrates the system behavior in a centralized fashion. On the other hand, fog nodes might be deployed in different tiers at the network edge to perform video streaming management, in order

to take advantage of VoD cached closer to avoid overloaded conditions of the network and video servers [8]. In this context, new services must be created to offer QoE-aware VoD services to mobile users, while optimizing the usage of heterogeneous network resources. For instance, VoD providers can run part of VoD services in the **cloud** in a more cost-effective fashion, but with higher values of latency for a stationary user with low QoE requirements. However, VoD providers migrate part of the VoD content to a **fog** node granting low latency with the high monetary cost for a mobile user with high QoE requirements. Hence, an efficient content orchestrator mechanism must periodically monitor network and user requirements, in order to select the most suitable node in the multi-tier **fog** environment for a user to access the VoD content, providing QoE assurance and efficient usage of network resources.

The content orchestrator works in two phases: (i) analysis and (ii) decision-making & execution. The analysis phase collects information from multi-tier **fog** nodes and QoE from the mobile user to understand their behaviors/availabilities to keep the QoE high, while optimizing usage of network and **cloud** resources. Decision-making & execution phase finds the best node in the multi-tier **fog** environment to distribute the VoD service for each user request by using a multi-criteria technique. This phase is also responsible for forwarding a content request to such **fog** node. Some investigations deal with content orchestrator mechanisms for VoD services [9, 10, 11, 12, 13, 14, 15]. However, network conditions continuously change over time. Previous works are limited to select a **fog** node only at the streaming startup and often neglect QoE information. It is mandatory for such mechanism to efficiently and periodically classify the best **fog** nodes to disseminate the content. This classification must consider different metrics to understand the performance of **fog** nodes and users to make efficient decisions. In this sense, it must consider the trade-offs between network conditions, QoE, and costs for decision making.

In this article, we introduce a content orchestrator mechanism for VoD streaming in a multi-tier **fog** computing environment, called Fog4Video [16]. We consider a network infrastructure to enable the cooperation between multi-tier **fog** and **cloud** nodes to meet user needs for VoD services. In this sense, we introduce a multi-tier **fog** architecture for VoD services with layered components used to implement Fog4Video in two phases. For the analysis phase, the Fog4Video collects information about available bandwidth, delay, stall duration, number of stall events, and monetary cost from the network, user client, and **fog** node. For the decision-making & execution phase, Fog4Video considers the Analytic Hierarchy Process (AHP)

method to assign different degrees of importance for each criterion. Therefore, Fog4Video selects the suitable fog node to stream VoD content cached on it, improving the QoE of delivered videos. Simulation results demonstrate the efficiency of Fog4Video in transmitting VoD compared to additional content orchestrator mechanisms. The number of stall events is reduced by up to 70%, and the stall duration is reduced by up to 65%. This behavior is an essential achievement of Fog4Video since stall duration and stall events cause the most negative impact on user perception than bitrate switching [17]. Fog4Video also improved the average bitrate by up to 35% and reduced monetary cost by up to 24% compared to other content orchestrator mechanisms.

The remainder of this article is organized as follows. Section 2 overviews the state-of-the-art for studies about a content orchestrator mechanism for VoD service. Section 3 describes Fog4Video, which is implemented in a multi-tier fog architecture. Section 4 details the simulation methodology and evaluation results of the Fog4Video mechanism. Finally, Section 5 presents the concluding remarks and future works.

2. Related Work

In this section, we introduce recent research on multi-tier fog computing systems to address the challenges in terms of Quality of Service (QoS) and QoE for VoD services. We also identify gaps in the literature, leading to design a content orchestrator mechanism in multi-layered and distributed fog systems. Byers et al. [6] propose a multi-tier architecture for several use cases, such as transportation, smart cities, and residential customers. The closer availability of content enhances delivery of video content in specific regions improving the network by reducing the network load and absolute latency. Khattak et al. [18] present a vehicular fog computing architecture for infotainment applications and evaluates in terms of cache size, cache hit ratio and energy of the vehicular fog nodes. Rosário et al. [19] describe the operational impacts and advantages regarding video content migration in a multi-tier architecture with QoE support. Iotti et al. [20] analyze the effect of proactive cache into fog nodes on the network edge to avoid redundant traffic. They classify and identify the manageable traffic based on DNS queries and replies. However, these works mostly provide optimizations lowering latency but lack in coordinating the content orchestrator to more efficient fog nodes in cases of poor QoE delivery.

Caching [21, 22] and Adaptive BitRate (ABR) [23] schemes aim at optimizing the QoE of delivered videos. Caching schemes enable users to access

popular content from caches placed near to the user [22]. ABR adapts the video bit rate according to the different network, device, QoE, and user characteristics [23, 12]. However, these video services are restricted to cloud and edge which can significantly increase the deployment cost compared to multi-tier. In this context, it is essential to explore how to combine the existing video services in each tier to improve QoE by dynamically manage the video content orchestrator to serving fog nodes with the available network, processing, storage, and cost of network devices [24, 25].

Several works introduced content orchestrator mechanisms exclusively at of request arrival. Tang et al. [9] redirect user requests to multiple destination servers to minimize a cost function considering service response times, computing costs, and routing costs. Siavoshani et al. [13] balance a load of storage resources and communication costs, managing the server redirection process based on cache size limitations and proximity to the server. Chunlin et al. [15] aim to minimize service time, power consumption, and costs from the service provider, leveraging multiple QoS parameters to select video service provisioning. However, these works are restricted to optimize QoS not necessarily improving QoE for VoD services. Moreover, they neglect overloaded network path after the selection of service provider. Xiao et al. [10] formulate geo-distributed and cloud-based dynamic content orchestrator redirection and resource provisioning as a stochastic optimization problem reducing costs for renting cloud resources to improve QoE by correlating with QoS metric of delay. Zhang et al. [11] propose content placement and request dispatching for cloud-based video distribution services with Markov decision process aiming to maximize profits of the video service provider considering cost and also correlating QoS with QoE. However, it is essential to consider QoE for VoD to provide more accurate information about the user's visual perception, which lacks when correlating network QoS metrics with QoE for VoD.

Based on our analysis of the state-of-the-art, we conclude that VoD services deployed in a multi-tier fog architecture improve the QoE by efficiently orchestrating fog nodes resources, while reducing delay and amount of data uploading/downloading to the cloud in a cost efficient fashion. Hence, it is essential to manage content requests during the entire streaming to avoid potential playback abandonment of users with poor QoE by streaming video from fog nodes with proper connectivity and resources. However, existing mechanisms still need to provide an efficient content orchestrator mechanism based on network, fog node, and user information to support VoD distribution with QoE support. Table 1 summarizes the main characteristics of previous works intended to provide VoD distribution.

Table 1: Summary of Characteristics of Video Content Orchestrator Mechanisms

	QoE	QoS	Request	Cost	Multi-tier
Iotti et. al. [20]		✓	✓		
Byers et. al. [6]					✓
Khattak et. al. [18]					✓
Rosário et. al. [19]		✓			✓
Wang, S et. al. [26]			✓		
He et. al. [27]					
Sheng et. al. [21]			✓		
Wang, X et. al. [22]			✓		✓
Pedersen et. al. [23]		✓			
Ge et. al. [12]		✓			
Abedini et. al. [28]		✓			
Tang et. al. [9]		✓	✓		
Xiao et. al. [10]		✓	✓	✓	
Zhang et. al. [11]			✓	✓	
Siovoshani et. al. [13]			✓		
Chunlin et. al. [15]		✓	✓	✓	

3. Fog4Video Content Orchestrator

In this section, we present the proposed Fog4Video content orchestrator. It aims to choose an appropriate fog node to distribute content throughout VoD streaming, considering network, fog node, and user's information for decision-making. Fog4Video performs the content orchestrator in a hierarchically network infrastructure, where multi-tier fog nodes could cache the content and also provide VoD services, improving the QoE of VoD service. The hierarchical design stands for performance, location, and cost deployment on each layer. Based on such architecture, Fog4Video classifies the connectivity and resources of each available fog node into a multi-criteria rank, where it considers the AHP method to assign different degrees of importance for each criterion to provide better QoE for each user. In the following, we introduce the multi-tier fog architecture, and also we describe the Fog4Video mechanism in detail as well as its deployment in such architecture.

3.1. Multi-tier Fog Architecture

Fog nodes can be deployed anywhere in a network, organized in tiers between the mobile devices (at the bottom) and the cloud (at the top) [19],

as it can be seen in Figure 1. The *Client Module* can be mobile devices consuming VoD content. The *Multi-tier Fog Module* can be any device in the Radio Access Network (RAN), e.g., Base Station (BS), or Access Point (AP) providing multimedia services to few dozens or hundreds of mobile devices. Based on the service demand, network path, and to sustain appropriate QoE, a replica of a tier can take place in the network, such as Base Band Unit (BBU) or Internet Service Provider (ISP). Besides, mobile devices could become a fog node to relay the video content via Device-to-Device (D2D) wireless communication for mobile devices with high and similar traffic demands. On top of such multi-tier architecture, there is the *Cloud Module* capable to provide VoD service in a centralized fashion keeping the entire database into a cloud computing datacenter.

Figure 1 shows the modules and components of the multi-tier fog architecture, which relies on two types of nodes: centralized cloud computing and distributed fog computing. Both work collaboratively to provide VoD services for client applications that aim to improve the QoE. Cloud nodes perform control functions, while fog nodes execute cache and VoD streaming services. The client application requests and displays the VoD content to users. In the following, we introduce the functionality of each component in the modules.

The *Client Module* consists of a *Client Agent* that manages communication among *Video Player* and *Client QoE/QoS Under Test*. The *Client Agent* plays the role of an interface between *Cloud* and *Client Modules*, synchronizing control, and data flow in both directions. Moreover, this module manages the migration of video streaming services to the fog node. The *Video Player* component downloads video content on the mobile device screen. While displaying video content, the *Client QoE/QoS Under Test* collects and reports QoE and QoS measurements, such as playback start time, stall duration, Mean Opinion Score (MOS), throughput, Round Trip Time (RTT), packet loss, and others, to understand QoE related to VoD.

The *Multi-tier Fog Module* includes a *Fog Agent* connecting the *Streaming Unit*, *Transcoding Unit*, *Cache Unit*, and *Fog QoE/QoS Under Test* components. The *Fog Agent* also plays the role of an interface between *Cloud* and *Multi-Tier Fog Modules*; synchronize control and data flow exchange in both directions; manage/provide communication among internal modules. *Transcoding Unit* runs on a fog node to adapt the video codec, bit rate, or resolution according to the network conditions, device capabilities, or QoE [23]. However, it can run only on a tier with sufficient resources available, since it requires more processing, data exchange, and memory capabilities. The *Cache Unit* stores redundant copies of given video content close to the

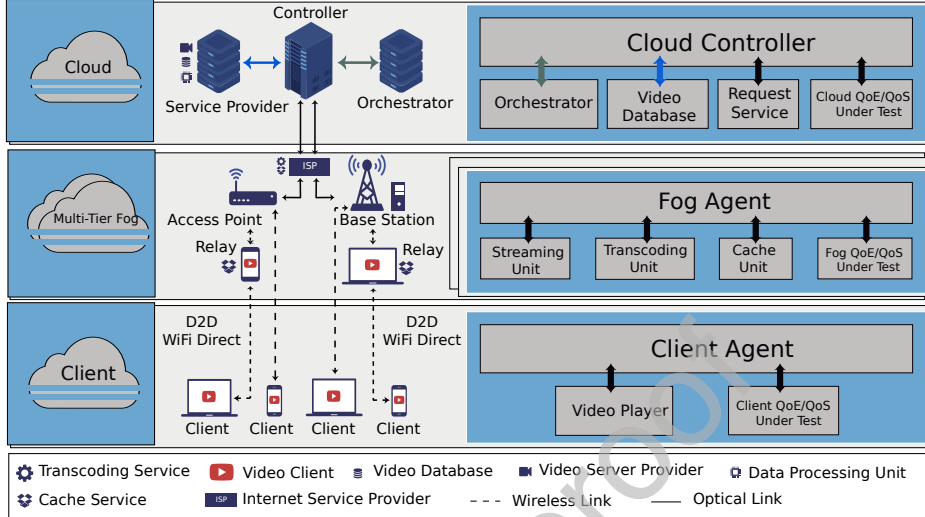


Figure 1: Multi-tier Fog Architecture for VoD Services

user. The *Streaming Unit* streams the content from *Cache Unit* for each *Video Player* request based on *Orchestrator* decisions. The *Fog QoE/QoS Under Test* collects QoE and QoS measurements and replays the demands made by the *Orchestrator*.

The *Cloud Module* consists of a *Cloud Controller* connecting the *Orchestrator*, *Video Database*, *Request Service*, and *Cloud QoE/QoS Under Test*. The *Orchestrator* coupled to the *Controller* steers decision-making on management and operation tasks based on the QoS/QoE reports from the clients. For example, the *Orchestrator* decides about fog deployment, service migration, evaluates available resources, and considers a specific content orchestrator mechanism to define where, what, and when a client must download the video from a different *Streaming Unit*. It holds input from the *Cloud QoE/QoS Under Test*, VoD requirements, and high-level management information, such as network-wide policies or Service Level Agreements (SLAs). The *Video Database* stores the VoD content for traditional DASH in the *Service Provider*, while the *Request Service* coupled to the *Controller* distributes the content to the *Video Player* at the client device. Moreover, it controls from which tier the *Video Player* must download the content, including when the download shifts from one fog node to another. The *Cloud QoE/QoS Under Test* built-in the *Controller* collects QoE and QoS measurements and replays the demands made by the *Orchestrator*. The *Cloud Controller* manages communication among internal modules, synchronizes

control and data flow exchange, and sends decisions taken by the *Orchestrator* to *fog* nodes. These decisions generate control flows to *Fog and Client Agents*. Hence, the controller of each tier can start the VoD service procedures to optimize the QoE of delivered videos.

In this architecture, the *Video Player* requests a video from the *Request service* at the *cloud*. Next, *Request Service* sends back the Media Presentation Descriptor (MPD), which is an XML file that contains information about the available video chunks, as it happens in DASH. It also includes other metadata needed by clients to choose between the available video chunks. In a VoD scenario, the video is divided into multiple chunks, and each chunk can be requested with a different bit rate representation to avoid buffer underflow, preventing stalling in varying network conditions. In this sense, the *video Player* requests the next chunk with an appropriate bit rate based on available transmission resources. In other words, the bit rate increases with sharpened network conditions and decreases in case of buffer underflow.

The *Client QoE/QoS Under Test* periodically sends QoE feedback, *i.e.*, the number of stalls and stalls duration values, to the *Orchestrator* unit. The stalls consist of interruptions on the video playback. Based on these values, Fog4Video implemented in the *Orchestrator* considers network, fog node, and user information to perform real-time content orchestration, *i.e.*, it chooses an appropriate *Streaming Unit* from a given tier for the client to download the video. The *Orchestrator* can also decide to keep downloading from the current *fog* node *Streaming Unit* as decided previously. Therefore, Fog4Video performs load balancing among *fog* nodes to better meet the user needs, *i.e.*, avoiding *Streaming Unit* overload while improving VoD distribution.

3.2. Orchestration

Fog4Video distinguishes two phases, namely, Analysis, and Decision-making & Execution. In the Analysis phase, it collects important metrics for decision-making, *i.e.*, available bandwidth, delay, number of stalls, stalls duration, and cost to deploy VoD services in a given tier. Afterwards, this information is evaluated in a Decision-making & Execution phase, which determines the best *Streaming Unit* to download the video. Finally, the *Cloud Controller* sends decisions taken by the *Orchestrator* to *fog* nodes and clients.

3.2.1. Analysis Phase

Cloud QoE/QoS Under Test, *Fog QoE/ QoS Under Test*, and *Client QoE/QoS Under Test* collect information from **cloud**, **fog** nodes, and the user, to understand their performance to make the best decision. Specifically, *Fog4Video* receives the QoS characteristics, *i.e.*, available bandwidth and delay, collected by *Cloud QoE/QoS Under Test* and *Fog QoE/QoS Under Test*, since these values impact QoE of VoD services. The VoD service uses TCP stream, which is highly affected by high latency in best-effort Internet [29]. As TCP, video adaptation algorithm based on TCP depends on quick feedback given by the clients. Therefore, the delay is essential to provide a more accurate response of the characteristics of the next chunk to be sent. The *Orchestrator* gives preference to idle and more cost efficient fog nodes before collecting QoS reports from the clients.

This phase also considers QoE metrics, *i.e.*, number of stall events, and stall duration, collected by *Client QoE/QoS Under Test*. Specifically, *Video Player* buffers the downloaded chunks before playing out and stops the video playback. As soon as the buffer level is empty, the video playback cannot continue, since there is insufficient data available in the buffer [2]. The interruption lasts until the fulfillment of a complete chunk in the buffer. These interruptions are stall events, and their duration is called stall duration. These two well-known objective metrics have the most crucial factor in QoE since they directly impact the continuity of the VoD session [2]. For instance, users who experience more interruptions in the video tend to watch the video for a shorter duration and are likely to be dissatisfied in the case of four or more interruptions [30]. Furthermore, viewers prefer a single but long stall event instead of several short stall events. Hence, not only the number of interruptions but also their duration affect QoE [31].

Fog4Video also considers $cost_k$ to process data in a given fog node k , which depends on the amount of CPU time the *Streaming Unit* or *Transcoding Unit* use CPU for processing P_k and the monetary cost per hour H_k . In this sense, a video content orchestrator should consider the trade-off between the increased cost and QoE of delivered VoD services. In general, the scale deployment of more centralized fog nodes tend to offer resources in a cost-effective manner per processing unit compared to more geographically distributed. The value $cost_k$ to stream the VoD service in a fog node is computed as follows.

$$cost_k = P_k \times H_k \quad (1)$$

Eq. 2 computes the overall cost C_k to process a video chunk to deploy

VoD services in a given fog node. It depends on $cost_k$ to process the bitrate representation r of a given video v and binary variable $\alpha_{v,k}^r$. A true value of $\alpha_{v,k}^r$ stands for the transcoding of a chunk b_v^r in a fog node k .

$$C_k = cost_k \times \sum_0^k \sum_0^v \sum_0^r b_v^r \alpha_{v,k}^r \quad (2)$$

Fog4Video checks the resource availability of the fog nodes, *i.e.*, the number of available computation resources to adapt video content accordingly to the video bitrate requested by the clients from a specific fog node. Table 2 summarizes the notations.

Table 2: List of main symbol used for Fog4Video mechanism

Notation	Description
k, v, r	Index for fog node, video, bit rate representation
P_k	Processing time usage in k -th fog node
H_k	Monetary cost per hour of k -th fog node
T_k	Resource availability per second of k -th fog node
b_v^r	Bitrate of the r -th representation of the v -th video
$\alpha_{v,k}^r$	Binary shows if k -th fog node transcoded the r -th representation of v -th video
$cost_k$	Cost to process data in the k -th fog node
C_k	Cost of video service provisioning in the k -th fog node
A_k	Resource demand in the k -th fog node
L	Candidate fog node list

3.2.2. Decision-making & Execution Phase

At this phase, the Fog4Video mechanism implemented in the *Orchestrator* at the cloud is responsible for selecting the best *Streaming Unit* of a given fog node from which the client should download the video. In the first step, the mechanism creates a list L of candidate fog nodes by checking the resource availability to compute content adaptation of each fog node to deploy the VoD service, as shown Algorithm 1. Fog4Video receives a chunk request in a bitrate $b_v^{r,t}$ at a given time t . Then, Fog4Video checks the resource demand A_k on time $t - 1$ based on $b_v^{r,t-1}$ and $\alpha_{v,k}^{r,t-1}$ for all fog nodes and videos. Fog4Video evaluates if the resource availability T_k can support the current resource demand in A_k for insertion of the fog node k in L .

From the list of candidate fog nodes, we consider network, fog node, and user metrics for decision making, which have different degrees of im-

Algorithm 1: Computing fog node candidate list

Input: $b_v^{r,t}, b_v^{r,t-1}, \alpha_{v,k}^{r,t-1}$
Output: L
Data: V, K, T_k

- 1 $A_k = 0$
- 2 **foreach** $k \in K$ **do**
- 3 **foreach** $v \in V$ **do**
- 4 $A_k = A_k + b_v^{r,t-1} \times \alpha_{v,k}^{r,t-1}$
- 5 **if** $T_k \geq A_k + b_v^{r,t}$ **then**
- 6 Push request $b_v^{r,t}$ for candidate fog node k into list L

portance on decision making. In this context, Fog4Video considers AHP [32] to compute the influence factor for each parameter. Specifically, AHP is a multi-criteria decision-making scheme capable of balancing inputs with different degrees of importance. AHP combines qualitative and quantitative elements for the analysis, allowing the system to find an ideal solution considering several metrics in the decision-making process. AHP recognizes a pairwise comparison between the numerical values of each parameter and their relative degrees of importance, to adjust their weights at runtime. As a result, a higher weight means higher importance for the corresponding criterion. The pairs must not contradict with each other, *e.g.*, if the metric i is twice more important compared to metric j , then j has $1/2$ importance than i . We consider seven importance levels to compare each pair of parameters, indicating how essential one parameter is compared to others, as shown in Table 3.

Table 3: Pairwise Importance Levels

$m_{i,j}$	Degrees of Importance
1	i is as important as j
2	i is slightly important than j
3	i is more important than j
4	i is much more important than j
$1/2$	i is slightly less important than j
$1/3$	i is less important than j
$1/4$	i is much less important than j

We consider a comparison matrix $A = M_{n \times n}$ with lines and columns

representing the metrics considered for decision-making to represent all pairwise comparisons. Variable n denotes the number of elements compared, as shown in Eq. 3. Each $m_{i,j}$ value in the matrix means how important the i -th element is compared to the j -th element. The degree of importance levels depends on subjective judgement related to abandonment rates of VoD due to poor QoE. We set the values to achieve higher improvements in terms of QoE, while also considering others metrics, unless stated otherwise.

$$A = M_{n \times n} = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{pmatrix} \quad (3)$$

For the metrics used by Fog4Video, we define the number of stalls as F , stall duration as E , delay as D , and cost for the deployment of VoD as C_k . The comparison matrix M indicates which parameters have higher priority than others, as shown in Eq. 4. For instance, in the first line, we see that the number of stalls F metric is twice more significant than stall duration E in the second line and three times more important than delay D in the third line. It is essential to highlight that if one criterion is considered to be twice more relevant than another, then the other is $1/2$ as important compared to the first. Note that the main diagonal of the matrix must always contain the value 1, as we compare a metric with itself.

$$M = \begin{matrix} & F & E & D & C_k \\ F & \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix} \\ E & \begin{pmatrix} 1/2 & 1 & 3/2 & 3 \end{pmatrix} \\ D & \begin{pmatrix} 1/3 & 2/3 & 1 & 2 \end{pmatrix} \\ C_k & \begin{pmatrix} 1/4 & 1/3 & 1/2 & 1 \end{pmatrix} \end{matrix} \quad (4)$$

AHP measures the influence factor $I_{i,k}$ assigning pairwise comparisons with the data on each fog node k . The influence factor is given by the sum of the multiplication of the current value of a metric $P_{i,k}$, *i.e.*, F , E , D , C_k , with the relative importance of the other metrics, as shown in Eq. 5. For example, if the values in P_k are $F = 1$, $E = 2$, $F = 15$ and $C_k = 1$, the influence factor of the delay metric would $15 \times (1 \times (1/3) + 2 \times (2/3) + 15 \times 1 + 1 \times 2)$, based on the third line of Eq. 4.

$$I_{i,k} = P_{i,k} \times \sum_{j=1}^n m_{i,j} \quad (5)$$

The influence factor of each metric serves as input for the score S_k of the current conditions in each fog node k , which is given in Eq. 6

$$S_k = \sum_{i=1}^n I_i \quad (6)$$

As each video may have different QoE requirements, we consider a weight matrix W to give different priorities for each video stream in V . In this sense, each column in W is the weight given by a video, as each type of video has different characteristics and needs specific management. In this case, various weights were assigned to each video type, as shown in Eq. 7.

$$W = (w_1 \quad w_2 \quad \cdots \quad w_v) \quad (7)$$

The decision matrix DM considers the combination of each video weight in W and score S_k of each fog node, based on Eq. 8

$$DM = \begin{pmatrix} w_1 * S_1 & w_1 * S_2 & \cdots & w_1 * S_k \\ \vdots & \vdots & \ddots & \vdots \\ w_v * S_1 & w_v * S_2 & \cdots & w_v * S_k \end{pmatrix} \quad (8)$$

The parameters in the matrix DM have a significant variation making a low accuracy analysis for the decision. To decrease the discrepancy between the values of DM , we perform a normalization in every parameter of DM using the arithmetic average \overline{DM}_k of the values of column k . The calculation gets the difference between a given tier and the average of all tiers, parameter by parameter, as shown in Eq. 9. In the end, we have the normalized matrix $\eta_{v,k}$ with the same dimensions of DM .

$$\eta_{v,k} = DM_{v,k} - \overline{DM}_k \quad (9)$$

Afterwards, we measure the Euclidean distance ξ between the attributes of the fog node chosen in $t - 1$ compared to the current conditions of the

other **fog** nodes within the overlapping regions $\eta_{v-1,k-1}$, based on Eq. 10. Considering the ξ value, we select the **fog** node with the highest value.

$$\xi = \sqrt{\sum_{k=1}^K (\eta_{v,k} - \eta_{v-1,k-1})^2} \quad (10)$$

Based on the higher value of ξ , the Fog4Video mechanism acknowledges the potential of a **fog** node to stream video content to the client meeting QoE requirements and considering the cost. In this sense, the Fog4Video mechanism informs its decision via the *Cloud Controller* to **fog** nodes and client, detailing about which *Streaming Unit* the client must request for the given chunk.

For the multi-tier scenario, each **fog** node embeds a set of modules and components to assist the content orchestrator. Fog4Video defines one phase for analysis and another for decision-making & execution to support dynamic content orchestrator in real-time. For the last phase, an AHP method balances the multi-criteria inputs and executes the decision-making.

4. Evaluation

This section describes the evaluation methodology, including scenario description, simulation parameters, metrics used to evaluate the QoE of delivered videos transmitted by different content orchestrator mechanisms. We define the scenario and simulation parameters in Section 4.1. We discuss the results and the findings of the proposal in Section 4.2.

4.1. Scenario description and methodology

We implemented Fog4Video, and the scenario, as shown in Figure 2, in the NS-3.29 simulator, and the implementation is available for download at [16]. NS-3.29 implements the protocol stack for communication between the mobile device and the network infrastructure to reach the VoD service provider. For the wired infrastructure, we considered the partial topology of the FIBRE testbed [19] to set delays of the long-haul communication between **fog** nodes and the **cloud**. We distributed the multi-tier **fog** nodes organized into three **fog** tiers and one **cloud** level. In tier 3, there is an edge server with an Wi-Fi AP in a local datacenter with resource availability of 10 Mbps for transcoding. In tiers 1 and 2, there is a regional data center with resource availability of 20 Mbps for transcoding, each. In the **cloud**, there is a powerful datacenter with a resource availability of 100 Mbps for

transcoding. Figure 2 shows the delay of long-haul communication between **cloud** and tiers. For the wireless infrastructure, we consider Wi-Fi 802.11n APs, channel bonding of 40 MHz in the center of a square area of 50 m^2 providing access to 40 randomly distributed mobile devices.

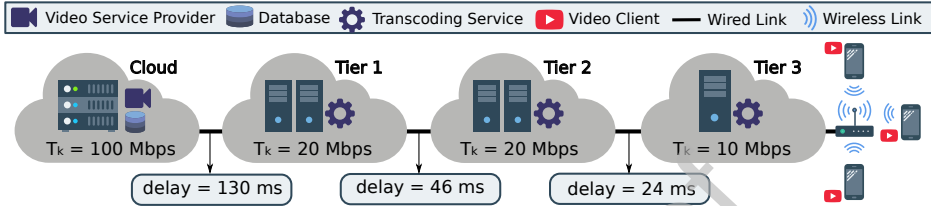


Figure 2: Multi-tier Fog Scenario

The mobile devices followed a linear continuous video request rate of 10 requests per second. By default, the video streaming initiates from the **cloud** node while the *Orchestrator* becomes aware of the QoE/QoS metrics of each client, such as stalling and delay. The adaptation algorithm is rate-based, where the bit rate starts from the lowest value and, for smoothing between each quality level, switches one level at a time following a conservative bit rate switching profile. The *Video Player* buffers at least 2000 ms, which is the size of a chunk. Therefore, as soon as the buffer does not have content to render (*i.e.*, stalling event), it has to re-buffer a complete chunk to play out the video again. We considered the Big Buck Bunny, Sunflower version video downloaded from the video library [33]. The *Video Player* at the client requests a video at a given time. Precisely, we use a High Definition video with a duration of 600 seconds, configured with 30 frames per second, and encoded into eight common used bitrates of 400, 650, 1000, 1500, 2250, 3400, 4700 and 6000 kbps [19], as shown in Table 4.

Table 4: Video Resolution and Bitrate Configurations

Resolution	180	360	360	540	540	720	1080	1080
Bit rate (Mbps)	400	650	1000	1500	2250	3400	4700	6000

QoE metrics overcome the limitation of QoS metrics to capture aspects of VoD related to the human perception [34]. In this way, we apply well-known QoE metrics for VoD services, namely bit rate, bit rate switch events, number, and duration of stalls [2]. Due to the conservative behavior of the adaptation algorithm, we consider the first 20 chunks as the *initial* bitrate since the first chunk always starts with the lowest bitrate, the last 20 chunks

for the *final* bitrate, and *average* bitrate of all the chunks. Depending on the content orchestrator, a *fog* node, chosen with the best QoE and cost improvement potential, can reply to most of the requests. We consider the Jain Fairness index F to express the concentration of requests [35] to measure requests fairness between *fog* nodes for each mechanism. The index calculation is denoted in Eq. 11, where x_i means the number of requests in *fog* node F_k .

$$F_k = \frac{[\sum_{i=1}^k (x_i)]^2}{k \sum_{i=1}^k (x_i)^2} \quad (11)$$

We also evaluate the cost C_k to deploy VoD services in a given *fog* node k , which is computed based on Eq. 2. The cost C_k for a given *fog* node k depends on the amount of time the *Streaming Unit* or *Transcoding Unit* uses resources for processing a chunk, causing a monetary cost M_k per hour of usage. In this way, we computed the monetary cost of CPU time per hour based on Amazon Web Services Cost of Ownership Calculator¹. The CPU time proportionally decreases when renting a higher number of CPU cores in the same AWS region. We considered the deployment into four regions and three Amazon instance setups and the monetary cost of each CPU core per hour, as well as memory and storage, are shown in Table 5.

We conducted 33 simulations for each of the three different *fog* content orchestrator mechanisms, namely, Random, Greedy, and Fog4Video. Then, we analyze their impact to deliver VoD content with QoE support and provide a 95% confidence interval. All mechanisms leverage the resource availability of the *fog* nodes. The Random strategy chooses the *fog* nodes with equal chances among all of them. The Greedy mechanism selects the *fog* node owning the smallest delay. Fog4Video evaluates the collected metrics to choose the best serving *fog* node, such as explained in Section 3.

Table 5: Monetary cost of CPU core time per hour based on Amazon ECS Setup

Node	CPU Cores	Memory	Storage	Cost
Cloud	40	160 GB	10 TB	\$0,07272
Tier 1	8	32 GB	2 TB	\$0,22896
Tier 2	8	32 GB	2 TB	\$0,22896
Tier 3	4	16 GB	1 TB	\$0,42408

¹<https://www.awstccalculator.com/>

4.2. Simulation Results

Figure 3 shows the number of clients per tier for each individual chunk, *i.e.*, tier 3, tier 2, tier 1, and cloud. This analysis provides information about the fog node selection behavior of each content orchestrator and from each tier the chunks were requested along the video playback. By analyzing the results, it is possible to conclude that all the clients start requesting the video from the cloud. Afterwards, each mechanism selects the *Streaming Unit* for the client to download the video in different ways. For instance, the Random strategy selects the tiers with a 25% probability, since it is one tier between four candidates. On the other hand, the Greedy mechanism prefers to select nodes with lower latency but picks the remaining tiers due to the resource availability in each tier.

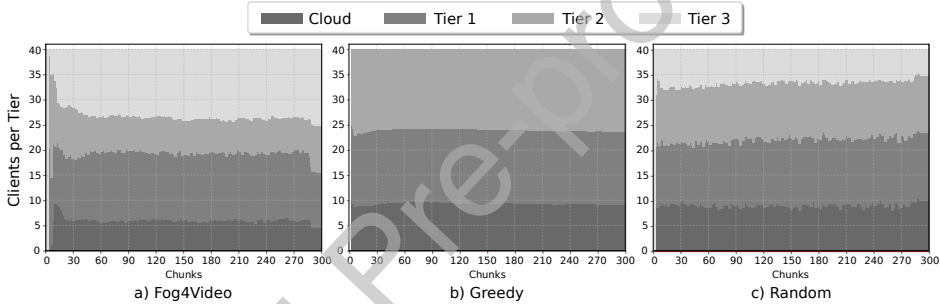


Figure 3: Number of clients requesting video chunks from each tier

Finally, Fog4Video selects the appropriate *Streaming Unit* based on network, fog node, user information, and cost, leading to higher use of tier 2 and the cloud. Tier 2 has the same cost as tier 1, but the delay is better in tier 2, leading to a more significant share. Tier 3 usage grows gradually before the 15th chunk because the bitrates of all clients are small in the beginning. When the bitrate grows, the clients move to other tiers capable of adapting the content as requested. In the last 15 chunks, the download of some clients finish, and other clients start to request from more cost effective tiers.

Figure 4 depicts the costs for VoD service deployment C_k for different content orchestrator mechanisms. By analyzing the costs, it is possible to conclude that Fog4Video reduces the cost by up to 24.04% and 16.32% compared to Greedy and Random, respectively. Fog4Video provides lower costs because it selects closer and more expensive tiers only when poor QoE is detected, despite the others mechanisms. On the other hand, Greedy and Random strategies do not follow this approach. In the last two cases, Greedy decides for the nodes with the lowest delays representing a closer

distance between **fog** nodes and clients. However, this decision incurs higher costs. The Random strategy has lower costs because of the lower number of clients requesting from the more expensive tier.

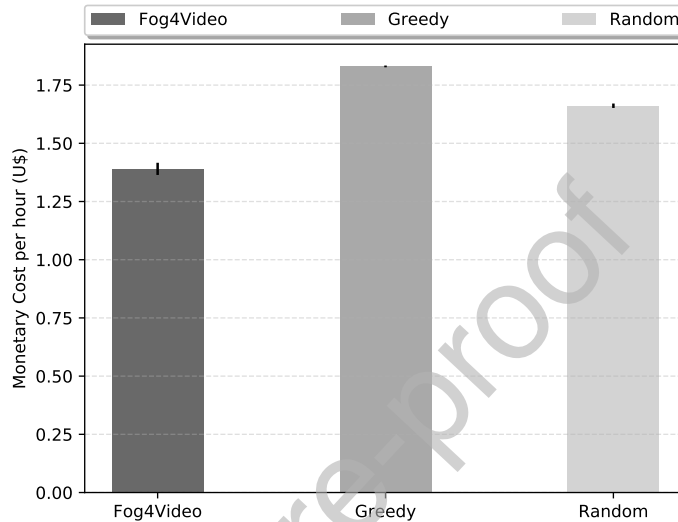


Figure 4: Cost to Deploy VoD Services

Figure 5 shows the bitrate *initial*, *final*, and *average* bitrates received by the client downloading the video via different content orchestrator mechanisms, *i.e.*, Fog4Video, Greedy, and Random. The video starts with a lower bitrate, *i.e.*, 400 kbps, regardless of the mechanism. We can also see that Fog4Video delivered the *final* bitrate up to 19.3% and 27.9% higher than Greedy and Random, respectively. The higher bitrate occurs because Fog4Video chooses the best *Streaming Unit* based on metrics like delay combined with QoE. The lower delay allows the adaptation algorithm to better predict the network conditions between the client and **fog** node with more frequent and updated information. Moreover, Fog4Video provides more fairness between the clients, giving a better opportunity to clients with worse QoE indicators, as shown in Figure 6. The fairness allows the clients to have more room to increase their bitrate and thus to have a better bitrate than the clients using Greedy and Random mechanisms. However, AHP needs around 15 chunks to evaluate the performance of each tier resulting in lower *initial* bitrate for Fog4Video. Finally, the *average* bitrate delivered by Fog4Video is 30.91% and 35.05% higher than provided by Greedy and Random mechanisms, respectively. The average is higher because Fog4Video has a short

period to adapt and converge. In this case, the clients can request better bitrates earlier, when compared to those clients using Greedy and Random, giving them a better overall bitrate.

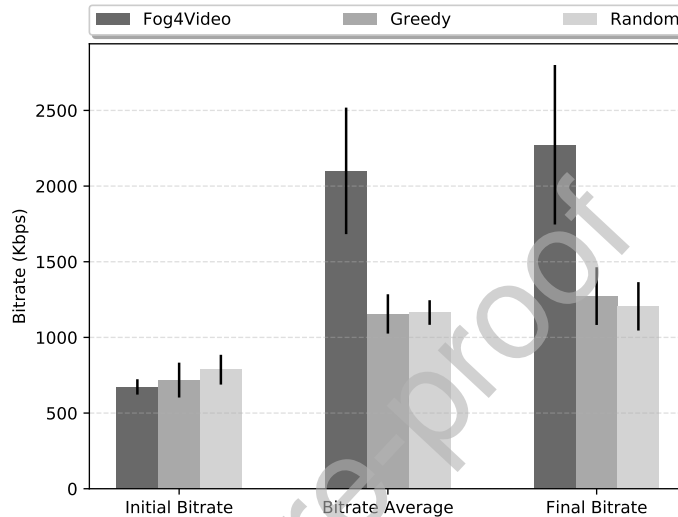


Figure 5: Impact of Content Orchestrator on Cost and Initial, Final, and Average Bitrate

Streaming Figure 6 shows Jain's Fairness Index for the client's distribution on fog nodes, which is computed by Eq. 11. The index shows the concentration of requests set by the mechanisms. The Random mechanism is the fairest because the probability to choose a tier is equal between all of them. However, this performance does not result in better QoE or cost-effective results by the mechanism. In this sense, Fog4Video offers the best trade-off between application performance and fairness, achieving a high score on the fairness index while cost-effectively improving QoE. Fog4Video decides to allocate requests to fog nodes with more potential to enhance QoE, and the fairness stands because of the usage of cheaper fog nodes. Moreover, the Greedy mechanism has a worse performance because it concentrates the requests to the closest servers even though it does not necessarily reflect in better QoE.

Figure 7 shows the number of stalls and their duration per client for videos delivered by each content orchestrator mechanism. Fog4Video reduces the number of stall events by 71.44% and 71.18% compared to Greedy and Random mechanisms, respectively. Moreover, Fog4Video reduces the duration of stall events by 72.45% and 65.23% compared to Greedy and Ran-

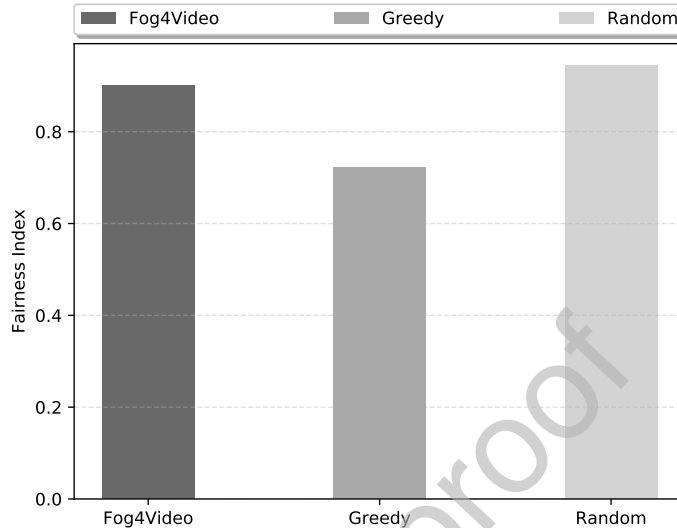


Figure 6: Fairness Index

dom mechanisms, respectively. These metrics have a significant influence on QoE, where high values could result in the viewer most likely leaving the video service. The interruption is a direct consequence of buffer starvation at the player, which is caused by poor network conditions, *i.e.*, long delay, between the client and the *Streaming Unit*. By analyzing the results, we can see that Fog4Video delivered videos for 40 clients with less than one stall per client. For example, around 16 clients experienced a single stall during the *Video Player*, which lasted about 0.68 seconds. The reduced number of stalls happens because Fog4Video proactively selects the best *Streaming Unit* based on network, fog node, and user information. The QoE metrics played an essential role in identifying how a fog node can potentially improve the user's satisfaction. On the other hand, the Greedy and Random mechanisms selected the *Streaming Unit* without considering such metrics, which do not avoid overloaded *Streaming Unit* for video delivery.

5. Conclusion

In this article, we introduced a multi-tier content orchestrator mechanism to provide QoE support for VoD service, called Fog4Video. It chooses an appropriate fog node considering network, fog node, and user information. Fog4Video is executed in fog nodes organized in multi-tier between

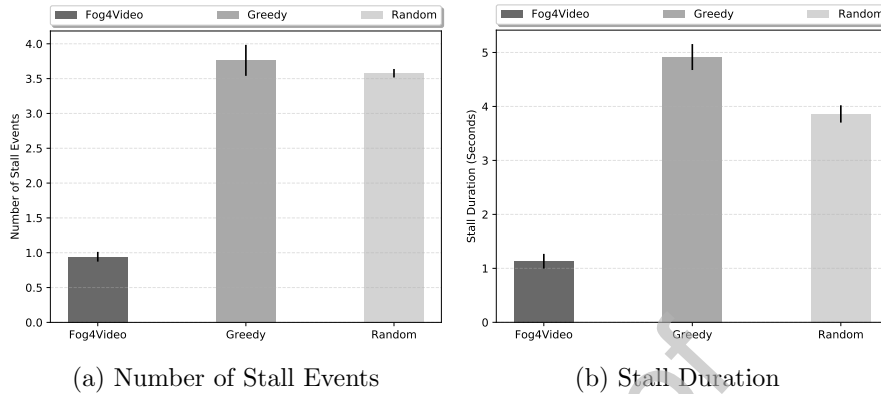


Figure 7: Impact of Content Orchestrator on Number and Duration of Stalls per Client

the cloud (at the top) and the mobile devices (at the bottom) to provide VoD services. Fog4Video provided better QoE regarding a VoD use case compared to delay based mechanisms. Fog4Video considers available delay, number of stalls, stalls duration, and cost to deploy VoD services in a given tier. The information of fog nodes and clients served as inputs for the AHP method to compute the influence factor for each parameter. From this, Fog4Video properly decided from which fog tier a better VoD provision with a lower overall cost can be achieved. From our evaluation analysis, we identified that Fog4Video delivered videos with an up to 30% QoE improvement compared to other content orchestrator mechanisms. The number of stall events reduces by up to 70%, and the stall duration reduced by up to 65%. These results are an essential achievement of Fog4Video since stall duration, and stall events significantly minimize the most detrimental factors that affect user perception. Fog4Video also improved average bitrate by up to 35% and reduced monetary cost by up to 24% compared to other content orchestrator mechanisms. Hence, simulation results show sufficient QoE assuring user's satisfaction in the use case of VoD.

For future works, the content orchestrator mechanism can consider the migration of the content to the fog nodes. This way, the prefetch of most popular videos and their representations would imply storage and transmission costs between caches and links of the fog nodes. These costs could guide the selection of the best fog nodes to cache more representations of the same video and avoid frequent transcoding. Besides saving transcoding costs, fewer redundant transmissions in the backhaul could provide better QoE. Moreover, we can consider additional QoS metrics to collect more

accurate information about network conditions, such as jitter and packet loss.

Acknowledgement

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. We also thank the National Council for Scientific and Technological Development (CNPq) for the financial support through Grant 431474/2016-8.

References

- [1] Cisco, Cisco Visual Networking Index: Forecast and Trends, 2017–2022, Technical Report, white paper, 2019.
- [2] P. Juluri, V. Tamarapalli, D. Medhi, Measurement of Quality of Experience of Video-on-Demand Services: A Survey, *IEEE Communications Surveys & Tutorials* 18 (2016) 401–418.
- [3] F. Araújo, et al., A hybrid energy-aware video bitrate adaptation algorithm for mobile networks., in: 15th IEEE/IFIP Wireless On-demand Network systems and Services Conference (WONS), pp. 146–153.
- [4] L. Reading, Optical/IP - Network operators brace for COVID-19 traffic spikes, Technical Report, white paper, 2020.
- [5] K. Bilal, et al., Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers, *Computer Networks* 130 (2018) 94–120.
- [6] Byers, Charles C, [Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks](#), *IEEE Communications Magazine* 55 (2017) 14–20.
- [7] A. Gomes, et al., Edge Caching with Mobility Prediction in Virtualized LTE Mobile Networks, *Future Generation Computer Systems* 70 (2017) 148–162.
- [8] Y. Mao, et al., A Survey on Mobile Edge Computing: The Communication Perspective, *IEEE Communications Surveys & Tutorials* (2017).
- [9] J. Tang, W. P. Tay, Y. Wen, Dynamic request redirection and elastic service scaling in cloud-centric media networks, *IEEE Transactions on Multimedia* 16 (2014) 1434–1445.
- [10] W. Xiao, et al., Dynamic request redirection and resource provisioning for cloud-based video services under heterogeneous environment, *IEEE Transactions on Parallel and Distributed Systems* 27 (2015) 1954–1967.

- [11] Z.-H. Zhang, X.-F. Jiang, H.-S. Xi, Optimal content placement and request dispatching for cloud-based video distribution services, *International Journal of Automation and Computing* 13 (2016) 529–540.
- [12] C. Ge, et al., Towards QoE-assured 4K video-on-demand delivery through mobile edge virtualization with adaptive prefetching, *IEEE Transactions on Multimedia* 19 (2017) 2222–2237.
- [13] M. J. Siavoshani, A. Pourmiri, S. P. Shariatpanahi, Storage, communication, and load balancing trade-off in distributed cache networks, *IEEE Transactions on Parallel and Distributed Systems* 29 (2017) 943–957.
- [14] A. Soltanian, D. Naboulsi, M. A. Salahuddin, R. Glitho, H. Elbiaze, C. Wette, Ads: Adaptive and dynamic scaling mechanism for multimedia conferencing services in the cloud, in: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, pp. 1–6.
- [15] L. Chunlin, et al., Optimal media service selection scheme for mobile users in mobile cloud, *Wireless Networks* 25 (2019) 3179–3192.
- [16] H. Santos, et al., Fog4Video Repository - available after publish acceptance, Technical Report, <https://github.com/huggosan/dash-migration>, 2020.
- [17] R. K. Mok, E. W. Chan, R. K. Chang, Measuring the quality of experience of http video streaming., in: *Integrated Network Management*, pp. 485–492.
- [18] H. A. Khattak, et al., [Integrating Fog Computing with VANETs: A Consumer Perspective](#), *IEEE Communications Standards Magazine* 3 (2019) 19–25.
- [19] D. Rosário, et al., [Service Migration from Cloud to Multi-tier Fog Nodes for Multimedia Dissemination with QoE Support](#), *Sensors* 18 (2018).
- [20] N. Iotti, et al., [Improving Quality of Experience in Future Wireless Access Networks through Fog Computing](#), *IEEE Internet Computing* 21 (2017) 26–33.
- [21] M. Sheng, et al., Video Delivery in Heterogenous CRANS: Architectures and Strategies, *IEEE Wireless Communications* 22 (2015) 14–21.
- [22] X. Wang, et al., Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems, *IEEE Communications Magazine* 52 (2014) 131–139.
- [23] H. Pedersen, S. Dey, Enhancing Mobile Video Capacity and Quality Using Rate Adaptation, RAN Caching and Processing, *IEEE/ACM Transactions on Networking* 24 (2016) 996–1010.
- [24] F. Jalali, et al., Fog computing may help to save energy in cloud computing, *IEEE Journal on Selected Areas in Communications* 34 (2016) 1728–1739.
- [25] R. Menchaca-Mendez, et al., Opportunistic mobile sensing in the fog, *Wireless Communications and Mobile Computing* 2018 (2018).

- [26] S. Wang, et al., Dynamic service migration in mobile edge-clouds, in: IFIP Networking Conference (IFIP Networking), 2015, pp. 1–9.
- [27] Q. He, et al., Fog-based transcoding for crowdsourced video livecast, *IEEE Communications Magazine* 55 (2017) 28–33.
- [28] N. Abedini, et al., Realtime Streaming with Guaranteed QoS over Wireless D2D Networks, in: Proceedings of the 14th ACM international symposium on Mobile ad hoc networking and computing, pp. 197–206.
- [29] P. Georgopoulos, et al., Using software defined networking to enhance the delivery of video-on-demand, *Computer Communications* 69 (2015) 79–87.
- [30] T. Hoffeld, et al., Initial delay vs. interruptions: Between the devil and the deep blue sea, in: Fourth International Workshop on Quality of Multimedia Experience, IEEE, pp. 1–6.
- [31] Y. Qi, M. Dai, The effect of frame freezing and frame skipping on video quality, in: International Conference on Intelligent Information Hiding and Multimedia, IEEE, pp. 423–426.
- [32] T. L. Saaty, Analytic hierarchy process, *Encyclopedia of Biostatistics* 1 (2005).
- [33] V. T. Library, Download Guide: Big Buck Bunny, Sunflower version, accessed date: Aug 2019. [online] <http://bbb3d.renderfarming.net/download.html>.
- [34] E. Liotou, et al., Quality of Experience Management in Mobile Cellular Networks: Key Issues and Design Challenges, *IEEE Communications Magazine* 53 (2015) 145–153.
- [35] R. K. Jain, et al., A quantitative measure of fairness and discrimination, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA (1984).

Author Statement

Hugo Santos: Conceptualization, Methodology, Software, Formal analysis, Data Curation, Writing - Original Draft

Derian Alencar: Software, Validation, Investigation, Visualization

Rodolfo Meneguette: Methodology, Software

Denis Rosário: Conceptualization, Writing - Review & Editing

Jéferson Nobre: Writing - Review & Editing

Cristiano Both: Writing - Review & Editing

Eduardo Cerqueira: Resources, Supervision

Torsten Braun: Writing - Review & Editing, Supervision

Hugo Santos received the graduation degree in Computer Engineering, in 2016, and Master degree in Electrical Engineering, in 2018, both from Federal University of Pará (UFPA), Brazil. He has started his Ph.D. Currently, he is under a jointly-supervision at the University of Bern, Switzerland. His research interests are in the field of fog computing, multi-access edge computing, vehicular networks, video-on-demand, virtual reality and quality of experience.

Derian Alencar start his graduation in Computer Engineering, in 2015, in Federal University of Pará (UFPA), Brazil. Currently, he is finishing his bachelor thesis. His research interests are in the field of fog computing, multi-access edge computing, vehicular networks, video-on-demand, virtual reality and quality of experience.

Rodolfo Meneguette is an Professor at Federal Technology Institute. He received his Bachelor's degree in Computer Science from the Paulista University (UNIP), Brazil, in 2006. He received his master's degree in 2009 from Universidade Federal de São Carlos (UFSCar). He received his doctorate from the University of Campinas (Unicamp), Brazil, in 2013. In 2017 he did his post-doctorate in the PARADISE Research Laboratory, University of Ottawa, Canada. His research interest are in the areas of vehicular networks, resources management, flow of mobility and vehicular clouds.

Denis Rosário received his Ph.D. degree in Electrical Engineering at the Federal University of Pará, Brazil with joint supervision undertaken by the Institute of Computer Science and Applied Mathematics of University of Bern, Switzerland in 2014. Currently, he is a Professor at Federal University of Par. His current research interests include the following topics: Multimedia adaptation, Wireless Networks, FANET, VANET, Mobility, Quality of Experience, and Software Defined Network.

Jéferson Campos Nobre is an Adjunct Professor at the Institute of Informatics, Federal University of Rio Grande do Sul (UFRGS). He holds a Ph.D. in Computer Science (UFRGS, 2015). Prof. Nobre has been involved in various research areas, mainly computer networks, network management, and computer system security. He has been publishing his works in important journals and conferences, such as the IEEE Communications Surveys and Tutorials, the IEEE Communications Magazine, the Journal of Network and Systems Management (Springer), the IFIP CNSM, IEEE/IFIP NOMS, and the IFIP/IEEE IM. He also served in the organization of scientific events and in the TPC of relevant symposia.

Cristiano Both is an associate professor of the Applied Computing Graduate Program at the University of Vale do Rio dos Sinos (UNISINOS), Brazil. He coordinators research projects funded by H2020 EU-

Brazil, CNPq, FAPERGS, and RNP. His research focuses on wireless networks, next-generation networks, softwarization and virtualization technologies for telecommunication network, and SDN-like solutions for the Internet of Things. He is participating in several Technical Programme and Organizing Committees for different worldwide conferences and congresses

Eduardo Cerqueira received his Ph.D. in Informatics Engineering from the University of Coimbra, Portugal (2008). He is an associate professor at the Faculty of Computer Engineering of the Federal University of Par (UFPA) in Brazil, as well as invited researcher at the Network Research Lab at UCLAUSA and Centre for Informatics and Systems of the University of Coimbra-Portugal. His publications include 5 edited books, 5 book chapters, 4 patents and over than 180 papers in national/international refereed journals/conferences. He has been serving as a Guest Editor for 6 special issues of various peerreviewed scholarly journals. His research involves Multimedia, Future Internet, Quality of Experience, Mobility and Ubiquitous Computing.

Torsten Braun got his Ph.D. degree from University of Karlsruhe (Germany) in 1993. From 1994 to 1995, he was a guest scientist at INRIA Sophia-Antipolis (France). From 1995 to 1997, he worked at the IBM European Networking Centre Heidelberg (Germany) as a project leader and senior consultant. Since 1998, he is a full professor of Computer Science at University of Bern. Currently, he holds an appointment of a vice president of the SWITCH (Swiss Research and Education Network Provider) Foundation since 2011. He was a Director of the Institute of Computer Science and Applied Mathematics at University of Bern between 2007 and 2011. He is serving as Deputy Dean of the Faculty of Science, University of Bern since 2017. He received best paper awards from LCN 2001, WWIC 2007, EE-LSDS 2013, WMNC 2014, and the ARMS-CC-2014 Workshop as well as the GI-KuVS Communications Software Award in 2009. In the scope of EU funded projects, he was leading WPs of FP6-EUQOS and FP7-MCN. Moreover, he coordinated national projects such as SNSF SwissSenseSynergy and SNSF CONTACT.



Cristiano



Jeferson



Denis



Rodolfo



Derian



Torsten



Eduardo



Hugo

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Journal Pre-proof