An Orchestrator Architecture for Multi-tier Edge/Cloud Video Streaming Services

Eduardo S. Gama[†], Natesha B V[‡], Roger Immich[§], and Luiz F. Bittencourt[†]
[†]Institute of Computing - State University of Campinas (UNICAMP), Brazil
[‡] Data Science and Computer Applications, MIT, MAHE, Manipal,India
[§]Federal University of Rio Grande do Norte (UFRN), Brazil
eduardogama@lrc.ic.unicamp.br, nateshbv18@gmail.com, roger@imd.ufrn.br, bit@ic.unicamp.br

***Abstract*—Video streaming has become a prevalent form of entertainment and a vital means of communication, but the challenges of delivering high quality video content over the internet are numerous. One of the key challenges is the varying network conditions that can significantly impact video streaming quality, such as bandwidth fluctuations and packet loss. To overcome these challenges, an adaptive video streaming architecture is needed to adjust the video streaming in real-time to match the changing network conditions and ensure a high Quality of Experience (QoE) for the end-user. This article presents MIGRATE, an orchestrator architecture for video streaming services capable of adapting to user demand in real-time. The study considers an edge/cloud multi-tier network infrastructure. In addition, an Integer Linear Programming (ILP) model and a Greedy solution are proposed to decide the distribution of connections between users and services. Experimental results show that based on the optimization strategy used, it is observed that there is a trade-off between the resources used and the QoE provided to users. Further, we discuss the importance of considering QoE metrics and user engagement in designing video streaming systems.**

***Index Terms*—Cloud computing, Edge computing, Quality of Experience, Video Streaming, Video-on-Demand**

I. INTRODUCTION

In many ways, the pandemic has impacted our social lives. The population has acquired a habit of consuming video streaming services on the top video content platforms. Content Providers have noticed a difference in their customers' behavior. The total time spent downloading and watching a plethora of multimedia content during the pandemic peak has increased substantially compared to the prior years [1]. To deliver this huge demand, the Over-the-top providers have adopted HTTP Adaptive Streaming solutions to attend to the end-users. Approximately 70% of all mobile traffic corresponds to multimedia data circulating over the Internet. In addition, the forecasts for the coming years still estimate consistent growth in this type of traffic [2].

Adaptive video streaming services are done through the HTTP Servers using the HTTP protocol requests/responses. Original videos are partitioned into segments of equivalent playback time on the server side, and each of these segments has multiple versions that vary in bitrate/resolution/quality [3]. On the user side, the player can request video segments sequentially and adjust the bitrate resolution dynamically to network conditions. The users' QoE is significantly affected by the segment video quality levels that the video player selects [4]. Considering video streaming platforms working over the network, the final mile of access networks is when QoE deterioration and resource limitations occur for consumers [5].

Edge computing can accommodate the new demands of cloud video traffic, leading to better QoE for users[6]. However, some level of planning is required to effectively manage edge resources. Firstly, the edge node selection problem, in which video streaming services may not be deployed on the best edge node. For example, after a handover in a multi-access edge scenario, the user may connect to a node logically distant from the deployed streaming services. Second, the content providers have to take advantage of the infrastructure flexibility. When the number of users grows, additional local nodes must be added to the distributed system, and network capacity is expanded to accommodate the neighborhood's growth. The system must aggregate the users on the proper edge servers capable of maintaining QoE guarantees. This factor shows that the hierarchy of multimedia content transmission must be scalable according to the pool of users [7], [8].

Therefore, content providers or network operators must find an appropriate edge node to meet their users' needs. With these questions to be tackled, a decision-making process for selecting nodes at the edge becomes evident, which must be done in real-time to maintain a good QoE and minimize the waste resources in edge nodes. Additionally, aggregating active users in a minimum number of edge nodes can help balance load traffic [9], [10]. The main contributions of this paper are presented below.

1) Introduce the **M**ult**I**-tiered ed**G**e/cloud o**R**chestrator **A**rchitecture for video s**T**r**E**aming (MIGRATE), which aims to perform the decision-making for video streaming services in real-time;
2) An ILP model and Greedy solution to decide the distribution of connections between users and services;
3) Analysis of the results shows that, depending on the optimization strategy used, there is a trade-off between resources and users' QoE.

The remaining article is organized as follows. In Section II, related works in video streaming and edge computing are discussed. MIGRATE is presented in Section III, describing the architecture and also the proposed optimization model. Section IV describes the simulation-based experimental setup, while Section V describes the results, which demonstrate the effectiveness of the proposed solution in improving QoE and balancing load traffic. Finally, Section VI summarizes the main findings and contributions of the research.

II. RELATED WORK

In this section, we examine previous research in the field of video streaming at the edge, specifically focusing on the models of orchestrators capable of managing services.

To ensure the best user experience, Farahani *et al.* suggested using an optimization model that employs a service approach with auxiliary caches [11]. Client requests are served by reverse proxy services placed at the edge and considered the shortest fetch time to serve users. Large-scale trials verify the accuracy of the proposed method, and the framework performance is compared to client-server approaches. The framework significantly improves the users' QoE. Bentaleb *et al.* [12], proposed a method based on a deep neural network to make predictions about the media that will be used. This method allows the edge server to anticipate demand for cloud-based video segments and make requests in advance. The experimental findings demonstrate a superior performance of the proposed solution over baseline models.

Shi *et al.* [13] discussed two edge challenges, namely the ping-pong effect and the edge node selection problem. To solve these challenges, a proposed edge node selection strategy considers the handover and the edge cache status. Further, the authors proposed a QoE-aware method to optimize QoE directly. Thus, the proposed policies are executed based on users' QoE to effectively utilize the cached content on the server BS as much as possible. Nguyen *et al.* [14], proposed an HTTP-based relay mechanism, where the throughput perceived by users is computed while the user receives the segments. This allows the server segment re-transmission during network fluctuations as long as better-quality segments are received. Additionally, the authors utilize the push property of HTTP/2 to decrease the number of requests. The mechanism has reduced the low-quality video time while also improving QoE.

The distribution of multimedia services in a hierarchical edge/cloud network also poses significant challenges. Santos *et al.* [15] addresses this problem by formulating the service allocation problem as an ILP, with the goal of reducing the number of network nodes while keeping latency in mind. The proposed approach strategically selects the node closest to the user to meet their demands and enhance the overall user experience. Santos *et al.* [16] presents an orchestration mechanism called Fog4Video, designed to select the edge nodes for downloading video content. The mechanism utilizes user feedback to evaluate video streaming from edge nodes. The work divides the edge into three layers to ensure storage capacity and thus improves the average bitrate and significantly reduces monetary costs.

Guan *et al.* [17], [18] demonstrate the performance of the tree-structure model and propose an algorithm with a hit rate improvement in the multimedia content and considerable memory consumption savings. The algorithm redirects a user's request for a video to the nearest cache. If the cache node does not receive the video, the edge node forwards it to the upper tier, which forwards the request to the upper tier until it reaches the source node.

Within the featured articles, only Shi [13] deals with aspects directly related to wireless network issues. The displacement of content to the network edge is not necessarily beneficial, so the connection between the cache and AP operates separately. However, dealing with a multi-level architecture brings aspects that can help or worsen network performance if not managed correctly. In summary, the previous approaches' efforts to improve users' QoE have focused on statically reducing traffic load or optimizing multimedia streaming.

Our work aims to bridge this gap by introducing an orchestrator MIGRATE for video streaming across multiple edge/cloud levels. MIGRATE addresses current challenges by implementing an orchestrator capable of adapting to real-time video demands. Additionally, we formulate an ILP model that considers the multi-tier edge network. The model aims to maximize infrastructure use without affecting the user's QoE.

III. MIGRATE: Multi-tiered edge/cloud orchestrator architecture for video streaming

This section presents an overview of the MIGRATE architecture. A formal scenario in multi-tier edge/cloud environments is described, which includes the different tiers of the environment. The detailed architecture components are then defined, including the different functionalities and responsibilities of each component.

A. System Overview

MIGRATE is an adaptive video streaming architecture that utilizes a multi-tier edge environment for efficient resource management and improved video streaming quality. It takes advantage of the interactivity between nodes at different tiers to make the best decision within the edge network. The nodes are accessible through various connection levels, ranging from specialized servers in the core network to micro-data centers in the radio-access network. As depicted in Figure 1, on the left-hand side, the edge can be divided into various tiers; The top tier consists of cloud servers. The three tiers of the edge network illustrate the hierarchical structure of the edge network. Tier 2 corresponds to the Core Network Regional Edge in this upper multi-level edge. The Access Network Edge in Tier 3 supports a few dozen to a few hundred local nodes on the edge. Tier 4 deploys Edge Gateways on locally hosted edge nodes with limited storage capacity.

The orchestrator can monitor the network conditions and allocate resources accordingly to ensure that video services are deployed on the most appropriate edge nodes. This hierarchical structure enables MIGRATE to make decisions tailored to each level's specific needs, ensuring efficient resource management and better video streaming quality for end-users. The right-hand side of Figure 1 shows the MIGRATE software architecture and the necessary interactions for the video streaming components.

B. Communication Component

In order to perform the communication operation between the network entities, the main Communication Component feature works as a control channel. All inter-entity communication is handled through a channel created by the main class. The design follows the DASH-IF specifications [19]. In this work, the communication functions are offered to redirect the user to an edge cache, and afterward, the execution of the optimization component, which involves sending messages through the communication protocol to the users. After receiving a server change notice, users can only communicate with a server that actively accepts them. In doing that, this switch server message must be sent for users to initiate requests on an already available server.

A communication flowchart is depicted in Figure 2. The client initiates the process by requesting content from a centralized node. The server, in turn, provides the client with a manifest file containing information about the communication component, their relationships, and any other data the client requires to select a server.

The manifest file also includes other metadata needed to select additional media segments from edge caches. Afterward,

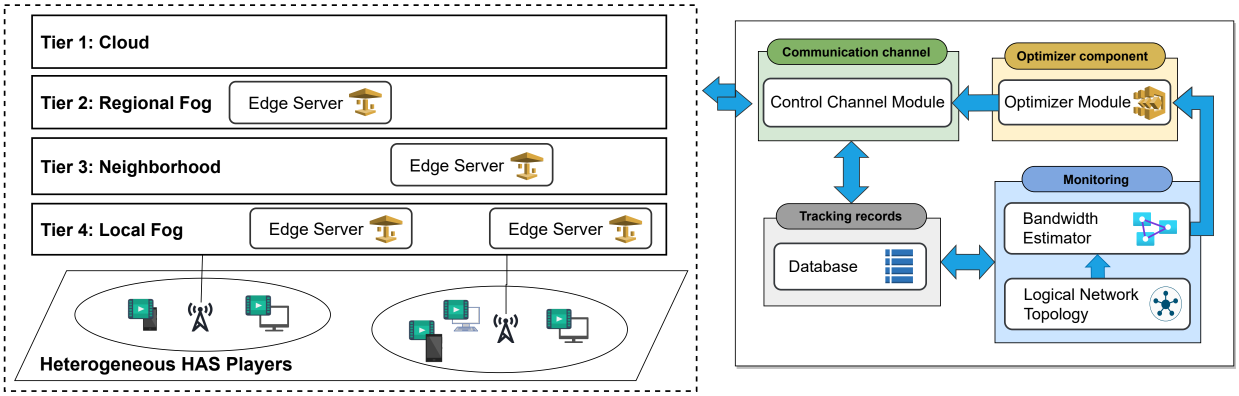


Fig. 1: Proposed MIGRATE orchestrator for a hierarchical environment.

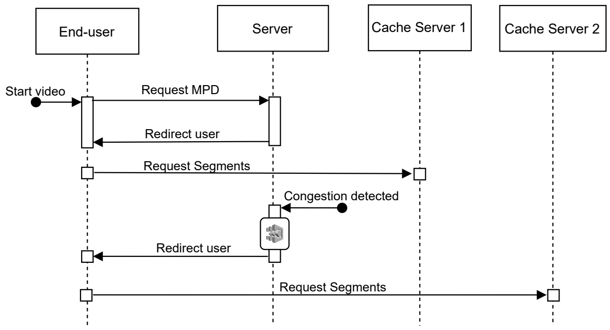


Fig. 2: Sequence Diagram of the video streaming communication.

the content is exchanged between the client and the cloud. In this context, user feedback sent to the cloud can trigger the service redirection to an edge node, e.g. due to QoE issues and link capacity. The data transmission can be provided both by the cloud and the edge nodes. Finally, the client can send additional feedback to the edge node that may lead to another service, either in a new or a pre-existing edge node in a different network tier or even in the cloud.

C. Monitoring Component

The monitoring component plays a central role in the overall system by observing predefined network metrics. These metrics are essential for the decision process of the MIGRATE orchestration, which must be aware of the network's real conditions to decide when a new edge node is needed. In this work, we will observe bandwidth consumption as a key metric. A network topology is represented by a non-directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of vertices is $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, where $\mathcal{V}_1$ represents the set of nodes capable of provisioning or serving in a backhaul topology of a mobile operator, and $\mathcal{V}_2$ is the set of viewers' locations. These vertices are connected by a set of edges $\mathcal{E}$, where each node $v_x \in \mathcal{V}_1$ has at least one edge $e_{xy} \in \mathcal{E}$ connecting to another node $v_y \in \mathcal{V}_1$, where $x, y = 1, 2, ..., |\mathcal{V}_1|, x \neq y$. Furthermore, V_2 represents the end-users u.

We consider a set of links with bandwidth ω_e, where $e \in \mathcal{E}$. Bandwidth estimation is based on the number of packets received and transmitted from network interfaces over a certain period t and for a given network status ψ. The estimated current link transfer rate is defined as $\omega_e(\psi, t)$. The total bandwidth capacity on the link e is represented by $\omega_e^T(\psi, t)$.

$$\omega_e(\psi, t) \leq \omega_e^T(\psi, t), \forall e \in \mathcal{E}, \forall\ t \tag{1}$$

Note that the estimated bandwidth $\omega_e(\psi, t)$ within the network should not exceed the total capacity $\omega_e^T(\psi, t)$ of each link e at any instant of time t, as given in Eq.(1). This is crucial to ensure that the network resources are used efficiently and that the end-users experience optimal performance.

D. Tracking Component

The tracking component registers the user's location, generating information on which the system can identify the best network management decision. When a streaming session is started, the Control Channel component first classifies the user into one of the groups based on specific characteristics of the manifest request and AP information. These operations can be done with the help of radio access level information received from the base stations, as well as the application level information that can aid in the whereabouts tracking in the video streaming system.

Here, we consider a set of users, where each user u is placed within a user group, with $\mathcal{V}_2 = \{g_1, g_2, ..., g_n\}$. In this way, a user's real-time location can be determined by the location area code (LAC), and the cell id $id(.)$, which we consider as a stopping point in where u is connected. We can combine these two fields to uniquely indicate the user's stay point as $g_i = \{LAC, \mathit{id}(u)\}$.

When a new user joins the video streaming system, the insertion algorithm adds the user to a group. Each set of users belonging to the same group is then aggregated into a single-node service. In this way, when a user requests video content, the multimedia server is able to route the user to a particular video provider rapidly. If the viewer is currently watching the video, a redirect message updates the server instead.

E. Optimization Component

The Optimization Component, which is in charge of making decisions regarding service placement and scalability, is a crucial part of the proposed architecture. These algorithms consider factors such as the current state of the network, the available resources, and the current services' workloads. As the demand for resources changes, the optimization component can quickly adjust the services to ensure that the network remains balanced and efficient.

Once the monitoring component detects a congested link, the optimizer component is triggered. First, the edge node selection module collects the input from the monitoring and repository. The first entry is a subset $\mathcal{F}_1 \subset \mathcal{V}_1$ of the network topology with nodes below the congested link. In doing so, a subset of user groups $\mathcal{F}_2 \subset \mathcal{V}_2$ are selected, since the selected groups g_i are connected to an Access Point $v \in \mathcal{F}_1$. For the sake of readability, the notations used throughout the paper are summarized in Table I.

TABLE I: Notation used in the proposed model

Notation	Description
ψ	Actual network state
$\mathcal{V}$	Set of all network nodes
$\mathcal{V}_i$	Subset of nodes i
$\mathcal{E}$	Set of links over nodes V
$\mathcal{F}_i$	Subset of nodes
$\mathcal{M}_j$	Total server capacity j
u	Current user u
g_i	Group of user i
p_j	Weight to serve the maximum number of users
x_{ij}	Binary variable for server selection
$\mathcal{NUM}_j(.)$	Number of user groups that pass through the same node j
$\mathcal{DEM}_e(.)$	Bandwidth required to attend a group's QoE at link e
$\mathcal{CAP}_j(.)$	Function that calculates the cache capacity of the video streaming in the edge server j

In a hierarchical model, the upper tiers within the network edge tend to serve more users. In order to maximize the number of users served by the same node, we used the function defined in Eq. (2). $\mathcal{NUM}_j(.)$ is a function that returns the total number of users who traverse node j. We aim to classify the best candidate for deploying the service based on p_j, defined as energy degree. Each node j holds its energy degree in accordance to $p_j = \mathcal{NUM}_j(\mathcal{F}_1)/\sum_{i\in\mathcal{F}_1}\mathcal{NUM}_i(\mathcal{F}_1)$.

This approach allows us to identify the best candidate nodes to deploy the services based on the energy degree. By deploying services on nodes with a higher energy degree, the same node will serve the maximum number of users. The Eq. 2 also seeks to limit the number of users through an upper limit. Thus, we can efficiently manage the limited resources available at the edge without affecting the end-users' QoE.

$$\chi(\psi, j) = \mathcal{NUM}_j(\mathcal{F}_1) * p_j \tag{2}$$

Eq. (3) represents the constraint that must be satisfied when assigning a video streaming service in an arbitrary node j. It guarantees that the assigned service is consistent with the available bandwidth ω_e. This constraint is essential to ensure that the users experience optimal performance and that the resources available at the network's edge are used efficiently.

$$\sum_{g_i\in\mathcal{F}_1} x_{ij} * \mathcal{DEM}_e(g_i) \leq \omega_e^T(\psi, t), e \in \mathcal{R}_{ij} \tag{3}$$

To support the video services demanded by users, an optimal route $\mathcal{R}_{ij}$ must be chosen between user group g_i and server j. The bandwidth required to attend the QoE of a users' group g_i is represented by $\mathcal{DEM}_e(.)$. If g_i requests to the server j x_{ij} is 1, otherwise 0.

With users accessing a video streaming from such a catalog and select the desired multimedia content, the demand for the video content segments can be approximately estimated, given the current audience. This allows efficient management in allocating cache resources to ensure that edge nodes are guaranteed continuous storage. Let s_i be an arbitrary multimedia content by a video server. The video streaming s_i starting at t_i as S_i^t (that is, $\mathcal{S}_i^T = \{s_i^{t_1}, s_i^{t_2}, ..., s_i^{t_n}\}$, where $(t_1, t_2, ..., t_n)$ are timestamps of videos $(s_1, s_2, ..., s_n)$ stored from t_i, respectively. The function $\mathcal{CAP}_j(.)$ calculates the cache of a given video stream, and the cache capacity of the edge server j, denoted by $\mathcal{M}_j$, is given by Eq.(4).

$$\sum_{g_i\in\mathcal{F}_2} x_{ij} * \mathcal{CAP}_j(S_i^T) \leq \mathcal{M}_j, j \in \mathcal{F}_1 \tag{4}$$

The proposed problem can be formulated as follows: To find the optimal set of candidates that maximize the energy degree while satisfying the necessary constraints. This can be achieved by maximizing the objective function in Eq 5.

$$\begin{aligned} \max \quad & \sum_{g_i\in\mathcal{F}_1}\sum_{j\in\mathcal{F}_2} \chi(\psi, j) * x_{ij} \\ \text{Subject to} \quad & (1), (3) \text{ and } (4) \end{aligned} \tag{5}$$

The proposed strategy considers the dynamic nature of network resources and adjusts in real-time to changing network circumstances, making it a resilient solution for managing multimedia services in edge-based networks. Once the ILP problem is solved, we receive the nodes on which the video streaming service should be deployed, as well as the connections between nodes $\mathcal{F}_1$ and $\mathcal{F}_2$.

IV. Performance evaluation

This section presents the performance evaluation of the MIGRATE orchestrator. The evaluation compares the results obtained from the ILP model two other models.

Figure 3 illustrates the scenario discussed in this work. We consider a cache hierarchy organized as a binary tree topology with seven nodes and a Cloud Provider connected to the root node. The four bottommost nodes are APs, and the others are edge nodes. The AP nodes are implemented in wireless devices that communicate via IEEE 802.11g at 2.4 GHz. The APs have wired connections to the edge nodes, while end users are connected wirelessly. Each user connected to the AP is located precisely 8 meters away from the AP. In order to force the use of edge nodes, the available bandwidth is 20Mbs on links 0-1 and 0-2 and 30Mbs on links 1-3, 1-4, 2-5, and 2-6. This configuration allows us to verify the spread of video streaming services at the edge. The bandwidth in $e_{0,1}$ and $e_{0,2}$ are smaller than in the lower links. In this way, the evaluated solutions can be used in the allocation of services with more efficiency in their performance in different conditions.

To implement the DASH servers and users that allow adaptive video streaming, we use the Adaptive Multimedia Streaming project (*Adaptive Multimedia Streaming* - AMuSt) [20]. The AMuSt framework provides a set of applications to produce and consume adaptive video based on the DASH standard. DASH functionalities are enabled by the libdash library, an open-source library that provides an interface to the DASH standard [21]. We consider that users are interested in a video available with ten different bitrate representations, namely

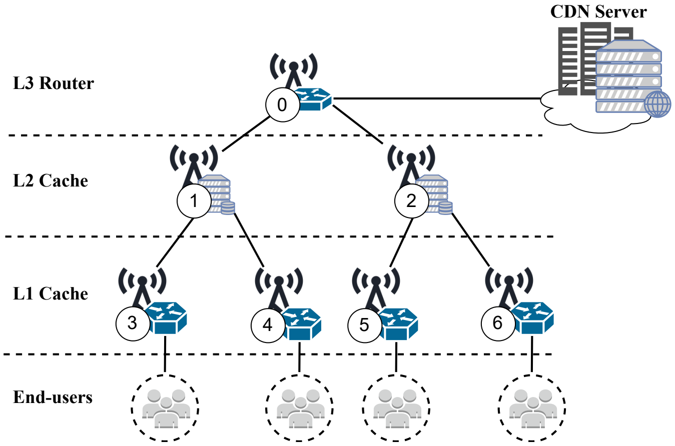


Fig. 3: Overview of the simulation topology.

{235kbps, 375kbps, 560kbps, 560kbps, 750kbps, 1050kbps, 1750kbps, 2350kbps, 3000kbps, 4300kbps, 5800kbps}, which are a subset used by Netflix [22]. Each representation is divided into 2-second segments. The number of streaming videos for download is limited to 10. The nodes' capacity is 10 in the cloud, while the edges are 9, 6, and 3 in levels 3, 2, and 1, respectively. A Zipf distribution with α equal to 0.7 is used to generate video selection and quality.

A. Decomposition Approach

We evaluate three scenarios: a *Cloud-Only* scenario without MIGRATE, and two scenarios using different strategies that seek to use the available resources with the MIGRATE orchestrator, called *QoS-Greedy* and *ILP Solution*.

The *Only-Cloud* scenario only uses the Cloud Provider node to deliver the video content. On the other hand, the *QoS-Greedy* approach uses the edge network nodes. Whenever link congestion is detected, *QoS-Greedy* is activated to choose which edge node to assist in delivering the video. The simulation's initial scenario starts with users requesting video content from the cloud. However, as soon as a congested link is detected, the edge caching mechanism below the congested link in the hierarchy is enabled, and the users receiving the video through the congested link are redirected to the activated edge nodes. This approach aims to alleviate the congestion on the network by reducing the amount of data that needs to be transferred over the congested link and, instead, serving the video content from the edge nodes located closer to the users. The third scenario, i.e. the *ILP Solution*, uses the proposed architecture, which aims to implement the video streaming service maximizing the use of the existing edge servers while respecting the user's QoE needs.

B. QoE Evaluation

Among the existing models in the literature, we describe how QoE metrics can be used to score user satisfaction. First, the video quality of each segment is calculated using a logarithmic law based on the bitrate [3]. The equation 6 shows the numerical transformation of the video quality received by the user. Each video has N segments and is encoded with L bitrate levels. r_i represents a specific bitrate level, and at each step i, the quality of the segment i is defined.

$$q(r_i) = a_1 * \log(a_2 * (r_i/r_L)) \tag{6}$$

To calculate the long-term satisfaction of each user, a flexible model is needed that includes the most influential metrics to quantify users' QoE. We consider Equation 7 [10], which encompasses four metrics: (a) the average perceptual quality of the chunk, (b) the average number of oscillations of quality, (c) the average number of stall events and their duration, and (d) the video startup delay. In Equation 7, K represents the total video segments, S_i is the duration of the stall, and ST_i is the startup delay of the user i.

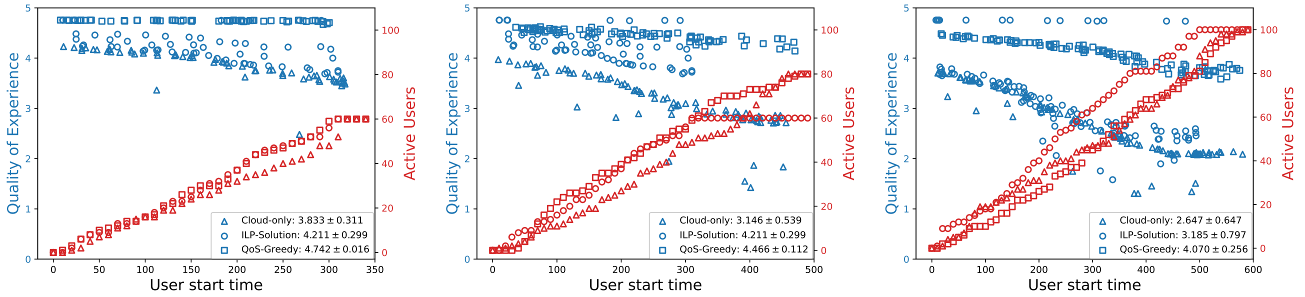
$$\begin{aligned} QoE_i = \frac{1}{K}\sum_{k=1}^{K} q(r_k) - \frac{1}{K-1}\sum_{k=1}^{K-1} |q(r_{k+1}) - q(r_k)| \\ - \frac{1}{K}\sum_{k=1}^{K} S_k - ST_i \end{aligned} \tag{7}$$

The QoE_i for each user i can range from 1 to 5, where 1 = bad, 2 = poor, 3 = fair, 4 = good, and 5 = excellent. By considering these different values, we can have a more comprehensive and accurate measure of the user's satisfaction.

V. Results

The experiments illustrated in Figures 4 (a), (b), and (c) present the average QoE, as calculated by Equation 7, for scenarios involving 15, 20, and 25 users per AP requesting videos in the simulated infrastructure. Each data point represents the overall QoE of each user in the *Cloud-Only*, *QoS-Greedy*, and *ILP Solution* scenario. The legends provide information on the standard deviation and mean values. The overall average performance of *QoS-Greedy* is better than the *Cloud-Only* and *ILP Solution* scenarios, mainly due to the choice of edge nodes to meet user requests. For example, in the *QoS-Greedy* experiment, when congestion occurs on intermediate links $e_{0,1}$ and $e_{0,2}$, the edge nodes closest to the users are activated. In this case, the first AP nodes would be activated just above the users. This way, the traffic going over the uplink will be smoothed so that users can have their QoE improved to an excellent level. Therefore, there is an increase in QoE in scenarios with the edge cache mechanism, as expected, given that users request segments from closer nodes. The performance difference in *QoS-Greedy* and *ILP Solution* scenarios for 15 and 20 users is approximately the same in satisfaction level. While for 25 users, there is a difference in the level of satisfaction.

An interesting discussion arises when QoE per user is analyzed separately, however. According to the numerical results, the final QoE tends to deteriorate as the number of active users increases. However, this is not entirely true for the *QoS-Greedy* scenario, where the final QoE for each user remains relatively high. The standard deviations of 0.016, 0.112, and 0.256, respectively, for scenarios with 15, 20, and 25 users per AP, suggest a relatively consistent QoE among users within the *QoS-Greedy* scenario. Only with 25 users per AP, the average QoE decreases with satisfaction close to regular, in contrast to the other two scenarios presenting good to excellent user satisfaction. In cloud-only scenarios, the network operates with a high standard deviation. As the number of users increases, some outliers appear with the lowest level of satisfaction,

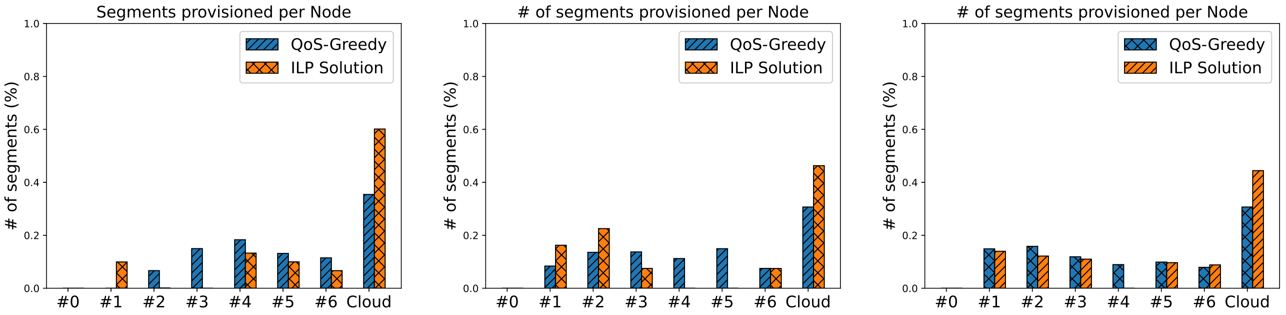


(a) 15 users

(b) 20 users

(c) 25 users

Fig. 4: Average QoE for each user. Number of users per access point in each scenario.



(a) 15 users

(b) 20 users

(c) 25 users

Fig. 5: Number of segments provisioned per node.

ranging between bad and poor. Finally, the Greedy algorithm's selection of edge nodes closest to the user effectively maintains a consistent QoE for users even when the number of active users increases. This highlights the effectiveness of the Greedy approach in ensuring a consistent and high level of user satisfaction.

Figure 5 (a), (b), and (c) illustrates the distribution of requests between edge servers. For the ILP solution, as the number of hops decreases, the number of requests between layers decreases for the 15 and 20 users scenarios. This can be understood by examining the formal model of the ILP, which only allocates new nodes for video streaming if the current edge servers do not meet the required demand. On the other hand, the QoS-Greedy approach always looks for the closest edge server possible to provision the video streaming, resulting in higher consumption of resources. As can be seen, a higher number of video streaming services are activated in the *QoS-Greedy* strategy when compared to the *ILP Solution*. This difference in resource usage is reflected in the request distribution between layers and servers, with the *QoS-Greedy* approach utilizing more resources to ensure a consistent and high level of QoE.

Based on these observations, the simulation results demonstrate the effectiveness of the proposed *QoS-Greedy* approach in providing high-quality video streaming services to users. The *QoS-Greedy* algorithm's selection of the closest edge nodes to the user helps to improve the QoE for users, and the standard deviation of QoE values is lower than for the other algorithms. On the other hand, the results of our analysis show that the *ILP solution* is more efficient in terms of resource usage when compared to the *QoS-Greedy* approach. As the number of hops decreases, the number of requests between layers decreases for the 15 and 20 users scenarios. This is due to the formal model of the ILP, which only allocates new nodes for video streaming if the current edge servers do not meet the required demand, whereas the QoS-Greedy approach always looks for the closest edge server possible to provision the video streaming, resulting in higher resource consumption. Therefore, a trade-off between quality and cost should be evaluated according to the expected demand and resource availability.

VI. Conclusion and Future Works

This article presents MIGRATE and investigates the characteristics of a multi-tiered Edge-Cloud scenario for video streaming services. The numerical results show that, depending on the optimization strategy used, there is a trade-off between the resources used and the QoE provided to the users. The ILP results indicate that by using a multi-layer Edge-Cloud architecture, it is possible to improve network resource utilization significantly. However, it negatively impacts user satisfaction due to bandwidth constraints. This highlights the importance of an effective management mechanism to optimize network resources and users' QoE depending on the users' demand.

Acknowledgment

This work was supported by the Innovation Center, Ericsson S.A., and by the Sao Paulo Research Foundation (FAPESP), grant 2021/00199-8, CPE SMARTNESS. This study was partially funded by CNPq and CAPES, Brazil - Finance Code 001.

References

[1] Roger Immich, Eduardo Cerqueira, and Marilia Curado. Efficient high-resolution video delivery over vanets. *Wireless Networks*, Feb 2018.

[2] Ericsson mobility reports, november 2022, 2022. Accessed 26-jan-2023.

[3] Weiwen Zhang, Yonggang Wen, Zhenzhong Chen, and Ashish Khisti. Qoe-driven cache management for http adaptive bit rate streaming over wireless networks. *IEEE Transactions on Multimedia*, 15(6):1431–1445, 2013.

[4] Roger Immich, Pedro Borges, Eduardo Cerqueira, and Marilia Curado. Qoe-driven video delivery improvement using packet loss prediction. *International Journal of Parallel, Emergent and Distributed Systems*, 30(6):478–493, 2015.

[5] Mukaddim Pathan, Ramesh K. Sitaraman, and Dom Robinson. *Advanced Content Delivery, Streaming, and Cloud Services*. Wiley Publishing, 1st edition, 2014.

[6] R. Immich, L. Villas, L. Bittencourt, and E. Madeira. Multi-tier edge-to-cloud architecture for adaptive video delivery. In *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 23–30, Aug 2019.

[7] Eduardo S. Gama, Lucas Otávio N. De Araújo, Roger Immich, and Luiz F. Bittencourt. Video streaming analysis in multi-tier edge-cloud networks. In *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 19–25, 2021.

[8] E. S. Gama, R. Immich, and L. F. Bittencourt. Towards a multi-tier fog/cloud architecture for video streaming. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pages 13–14, Dec 2018.

[9] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos. A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 20(1):416–464, Firstquarter 2018.

[10] Abdelhak Bentaleb, Ali C. Begen, Saad Harous, and Roger Zimmermann. Want to play dash?: A game theoretic approach for adaptive streaming over http. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, pages 13–26, New York, NY, USA, 2018. ACM.

[11] Reza Farahani, Farzad Tashtarian, Alireza Erfanian, Christian Timmerer, Mohammad Ghanbari, and Hermann Hellwagner. Es-has: An edge- and sdn-assisted framework for http adaptive video streaming. In *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '21, page 50–57, New York, NY, USA, 2021. Association for Computing Machinery.

[12] Muhammad Jalal Khan, Abdelhak Bentaleb, and Saad Harous. Can accurate future bandwidth prediction improve volumetric video streaming experience? In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1041–1047, 2021.

[13] Wanxin Shi, Qing Li, Ruishan Zhang, Gengbiao Shen, Yong Jiang, Zhenhui Yuan, and Gabriel-Miro Muntean. *QoE Ready to Respond: A QoE-Aware MEC Selection Scheme for DASH-Based Adaptive Video Streaming to Mobile Users*, page 4016–4024. Association for Computing Machinery, New York, NY, USA, 2021.

[14] Minh Nguyen, Christian Timmerer, and Hermann Hellwagner. H2br: An http/2-based retransmission technique to improve the qoe of adaptive video streaming. In *Proceedings of the 25th ACM Workshop on Packet Video*, PV '20, page 1–7, New York, NY, USA, 2020. Association for Computing Machinery.

[15] Fillipe Santos, Roger Immich, and Edmundo Madeira. Multimedia microservice placement in hierarchical multi-tier cloud-to-fog networks. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1044–1049, 2021.

[16] Hugo Santos, Derian Alencar, Rodolfo Meneguette, Denis Rosário, Jéferson Nobre, Cristiano Both, Eduardo Cerqueira, and Torsten Braun. A multi-tier fog content orchestrator mechanism with quality of experience support. *Computer Networks*, 177:107288, 2020.

[17] Yu Guan, Xinggong Zhang, and Zongming Guo. Caca: Learning-based content-aware cache admission for video content in edge caching. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 456–464, New York, NY, USA, 2019. ACM.

[18] Yu Guan, Xinggong Zhang, and Zongming Guo. Prefcache: Edge cache admission with user preference learning for video content distribution. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4):1618–1631, 2021.

[19] Dash-if candidate technical specification: Content steering for dash, 2023. Accessed 28-feb-2023.

[20] Christian Kreuzberger, Daniel Posch, and Hermann Hellwagner. Amust framework - adaptive multimedia streaming simulation framework for ns-3 and ndnsim, 2016.

[21] C. Mueller, S. Lederer, J. Poecher, and C. Timmerer. Demo paper: Libdash - an open source software library for the mpeg-dash standard. In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–2, July 2013.

[22] Per-title encode optimization, 2015. https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2; accessed 20-novembro-2019.