



**HAL**  
open science

# Combining Resource-Aware Recommendation and Caching in the Era of MEC for Improving the Experience of Video Streaming Users

Ana Claudia B. L. Monção, Sand Luz Correa, Aline Carneiro Viana, Kleber Vieira  
Cardoso

## ► To cite this version:

Ana Claudia B. L. Monção, Sand Luz Correa, Aline Carneiro Viana, Kleber Vieira Cardoso. Combining Resource-Aware Recommendation and Caching in the Era of MEC for Improving the Experience of Video Streaming Users. IEEE Transactions on Services Computing, 2022, pp.1-14. <10.1109/TSC.2022.3205482>. <hal-03996912>

**HAL Id: hal-03996912**

**<https://inria.hal.science/hal-03996912v1>**

Submitted on 20 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Combining Resource-Aware Recommendation and Caching in the Era of MEC for Improving the Experience of Video Streaming Users

Ana Claudia B. L. Monção, Sand Luz Correa, Aline Carneiro Viana, and Kleber Vieira Cardoso

**Abstract**—The coupling between content caching at the wireless network edge and video recommendation systems has shown promising results to optimize the cache hit and improve the user quality of experience (QoE). However, the quality of the UE wireless link and the resource capabilities of the UE are aspects that impact user QoE and that have been neglected in the literature. In this work, we present a resource-aware optimization model for the joint task of caching and recommending videos to mobile users that maximizes the cache hit ratio and the user QoE under the constraints of UE capabilities and the availability of network resources. In order to make the problem manageable, we assume that the regular user consumes video content keeping some time interval between them, and this user moves slowly inside the coverage of a base station. We evaluate our proposal using a video catalog derived from a real-world video content dataset and real-world video representations and compare the performance with a state-of-the-art caching and recommendation method unaware of computing and network resources. Results show that our approach increases user QoE by at least 68% and cache hit ratio by at least 14% in comparison with the other method.

**Index Terms**—video caching, recommendation systems, multi-access edge computing, quality of experience

## 1 INTRODUCTION

ACCORDING to forecasts presented by Cisco, IP traffic will increase more than three times by 2023, while video will be 82% of all this traffic [1], with a large part of such demand being generated by mobile devices to access video streaming. In this context, it is essential to adopt solutions that take into consideration the quality of experience observed by users and also the availability of resources in the cache, the users' devices, and the network. Recently, the joint use of content caching at the wireless network edge and recommendation systems have exhibited promising results [2], [3].

Content caching at the network edge is not new. Indeed, this idea was introduced by Content Delivery Networks (CDNs) more than 20 years ago. The Multi-access Edge Computing (MEC) paradigm [4], however, introduces some new features to this context. First, MEC provides the opportunity for mobile network providers to offer computing resources at the network edge, which may be also available to service providers. Second, MEC is not limited to offering the storage capacity and basic processing of classic CDNs, it can provide advanced processing, similar to those found in cloud systems, close to end-users [5]. Finally, the MEC platform can provide network status information (e.g., wireless channel conditions, traffic patterns, and user mobility patterns), which can be used to offer better services for users [6]. Thus, MEC is designed not only to allow the execution of traditional services such as video caching, but also new advanced services such as contextualized content, fine-

grained recommendation, virtual/augmented/mixed reality, autonomous cars, and industrial automation.

To serve different services and applications, mobile network providers need to deploy or integrate MEC resources in different parts of their networks, such as collocated with the Base Station (BS), collocated with some device of the Radio Access Network (RAN), or collocated with the Core Network (CN) functions (i.e., in the same data center) [5], [7]. Fig. 1 illustrates the last approach (i.e., MEC collocated with the CN functions), which is satisfactory for video streaming services (e.g., Netflix, Hulu). In general, the latency from the users until the CN is negligible for video streaming services, and the main bottleneck (i.e., Internet access) can be avoided if the content is cached. In fact, this is a common property of *on-demand* video streaming services which is the type of video service generally coupled with a recommendation system. Thus, it is out of scope in this work *real-time* video streaming services, which have different properties. Other properties of *on-demand* video streaming services are: the traffic is predominantly downlink, multiple representations of video content are available in the cloud, and several content caches are deployed inside the network of Internet service providers. Such caches are usually reachable within six IP hops and may increase throughput by a factor of three for 75% of the accesses [8].

On the other hand, recommendation systems, whose goal is to recommend content that matches the preferences of individual users, have become fundamental components of on-demand content delivery services. As an example, Netflix has reported that 80% of the hours streamed by the company comes from recommendations made by its system [9]. This fact illustrates how recommendation systems influence user accesses to content and, thus, they can be employed to induce access patterns that improve the cache

- Ana Claudia B. L. Monção, Sand Luz Correa and Kleber Vieira Cardoso are with Instituto de Informática, Universidade Federal de Goiás, Goiânia, Brasil.  
E-mail: anaclaudia@inf.ufg.br, sand@inf.ufg.br, kleber@inf.ufg.br
- Aline Carneiro Viana is with Inria, France. E-mail: aline.viana@inria.fr

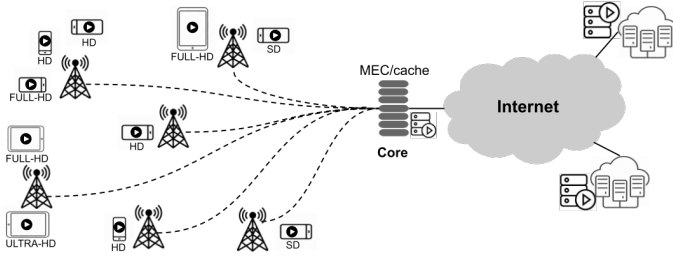


Fig. 1. Reference scenario for the system model.

performance. In [3], the authors elaborate on this idea and present an optimization model, denoted Joint Caching and Recommendation Problem (JCRP). JCRP combines caching and video content recommendations to maximize the cache hit ratio, while minimally affecting user preferences. The recommendation system is used to shape the user demand towards content that can be shared by multiple users, resulting in an improved cache hit ratio and also user quality of experience (QoE).

Nevertheless, caching and recommending popular content at the network edge is not enough to ensure the user QoE. Context factors such as the capabilities of the user equipment (UE) to reproduce the video and the quality of its wireless link may also affect the user satisfaction [10]. Thus, videos with resolutions that the user mobile device (UE) cannot reproduce should not be among her recommendations, even if the content is of her interest. Similarly, videos encoded with representations that require more throughput than the present user wireless link bandwidth should not be included in the user recommendations. Therefore, information on the UE capabilities and the quality of the wireless link by the time of the video recommendation is important to provide a better service to users. Despite these facts, no previous work has tackled the problem of caching and recommending videos in mobile networks taking into account content preferences and the availability of the computing resources in the UE and the quality of its wireless link.

In this work we propose an optimization model, denoted *Resource-Aware Video Recommendation* (RAViR), for the joint task of caching and recommending videos to mobile users that consume on-demand video streaming services. Our model takes into consideration content preferences and information on the availability of computing and network resources (provided by the MEC platform) by the time of the video recommendation. Consider, for example, an on-demand streaming service composed of a video cache service deployed in the CN and serving mobile users with different UE capabilities and under different wireless link conditions, as illustrated in Fig. 1. The general idea is that the recommendation system of the streaming service should recommend to individual users videos that i) match properly their content preferences; ii) provide the best QoE according to the UE capabilities and the quality of the wireless link by the time of the recommendation; and iii) have the potential to be demanded from multiple users. Our model is formulated under two assumptions. First, we assume that the regular user moves slowly inside the coverage of a base station. Consequently, while short-term changes in the wireless link are expected, the long-term average of the link quality tends to be stable for this type of user. We recognize that this assumption limits the expected

mobility scenarios for a 5G network. However, our model is still relevant, as no previous work has considered the interplay between cache, user preferences, and context information. Second, we assume that the regular user consumes video content keeping some time interval between them. As a result, users' content request patterns change slowly over time. This last assumption is usually taken in the literature related to on-demand video streaming services (e.g., [3]) since the large majority of the content available in the catalog of such services are recordings of series, television shows, and movies, whose duration usually is at least 30 minutes. To illustrate, in the dataset we use in our experimental evaluation (Section 5), more than 97% of the videos are longer than 60 minutes. This paper advances the state-of-the-art making the following contributions:

- We formulate and solve an optimization problem for the joint task of caching and recommending videos to mobile users, considering content preferences and the availability of computing and network resources. The objective of RAViR is to recommend videos to users that maximize the cache hit ratio and the user QoE (concerning content preferences and video representations) under the constraints of UE capabilities and the availability of network resources by the time of the recommendation.
- We provide an extensive evaluation of RAViR using a real dataset from the MovieLens project [11] and compare the obtained results with different variants of the JCRP [3]. Results show that RAViR increases user QoE by at least 68% and effective cache hit ratio by at least 14% in comparison with the JCRP variants.
- We introduce a new metric to assess the QoE of mobile users that consume on-demand video streaming services in the era of MEC. Our QoE metric is designed to capture the capabilities of the UE as well as the effect of the availability of computing and network resources on the user experience.

The remainder of this paper is organized as follows. In Section 2, we discuss the related work. The system model and the problem formulation are presented in Sections 3 and 4, respectively. We evaluate the performance of RAViR in Section 5. Section 6 presents conclusion and future work.

## 2 RELATED WORK

Caching is a classical subject in computer science and has been well studied in computer networks [12], [13]. However, the expected wide adoption of MEC in 5G networks is motivating the research community to revisit the topic under a refreshed perspective. This section briefly presents an overview of related research on caching and recommendation systems as well as QoE in video streaming services in the context of 5G networks.

### 2.1 Caching and Recommendation Systems

Few works in the literature have based caching decisions on personalized recommendations issued by recommendation systems [3], [14]–[17]. From this group, the first four deals with video content, and only [16] and [3] tackled the problem in the context of the network edge. The authors

in [16] introduce the concept of “soft cache hits” in the context of entertainment-oriented content consumption on the Internet. The idea is to recommend similar cached content to mobile users that request videos that are not cached at the BSs. Since the majority of video content on the Internet is entertainment-oriented, the authors argue that users are likely to accept the recommendations.

The authors in [3] evolve the initial idea introduced in [16] and propose a model to tackle the Joint Caching and Recommendation Problem (JCRP) in small cell networks. In JCRP, the recommendation system is used not only as a predictor of video content demand but also as a demand-shaping tool to enhance cache performance. To this end, the JCRP model issues recommendations that may not necessarily rank top in the inferred user content preferences but still score high in them. By bounding the distortion that such recommendations introduce to the original user content preferences, the JCRP model controllably guides user demand toward content that attracts the preference from multiple users. The authors show that using recommendation systems as a traffic engineering tool brings relevant gains in cache performance without significantly degrading the user preferences. However, in [3] user preferences are defined only in terms of content. Video resolution and the computing and network resources available at the user side to play the video are not taken into consideration. As a result, JCRP may recommend videos which representations are not suitable for the available user resources. In addition, although the cache is assumed to be collocated with multiple BSs, the authors do not describe how this distributed cache is managed, which is a non-trivial problem. Thus, while our work starts from a basis similar to [3], we make some different assumptions and employ context information (i.e., UE capabilities and network resource) that has a relevant impact on the final results. First, differently than [3], we consider centralized cache management in each RAN, which is the traditionally adopted approach for on-demand video streaming and avoids several issues concerning distributed cache management. Second, video resolution is not discussed in [3], but this property severely affects cache occupancy and so the cache hit. Our proposal is designed to take into consideration not only video resolution but also its impact on cache occupancy, and device and network resources. In fact, these resources are not taken into consideration in [3], making the approach unaware of the effective impact of its choices. In [3], users’ QoE is associated only with the cache hit ratio, which is not enough to effectively assess the users’ experience. We improve this evaluation by including the additional information in the QoE metric.

## 2.2 QoE in Video Streaming Services

The work in [3] assesses the user QoE using only the cache hit ratio. However, the cache hit ratio is not enough to measure if the video content is being delivered to the users with the appropriate quality of service (QoS). Indeed, many papers have investigated how to measure or predict the QoE of mobile users when they consume on-demand video streaming services. For example, a work from Netflix [18] proposes different ways to assess the user QoE by combining images and mobility metrics. Most of these metrics compare different video encodings with evaluations

issued by the users. In [19], the authors propose a low-complexity network metric to derive the QoE perceived by users of UHD video flows in 5G networks. This metric, denoted Congestion Index (CI), represents the ability of the network to successfully deliver video stream based on the maximum available bandwidth on the path from the server to the user. The QoE is then obtained using a regression technique that correlates the CIs (resulting from variable video bitrates and available bandwidth) with subjective user assessments. Indeed, most of the QoE metrics for on-demand video streaming services proposed in the literature are based solely on video transmission parameters (e.g., video bitrates and network throughput). Differently, in our work, we first propose a new Congestion Index (CI) that assesses the ability of the network to deliver video stream when the video comes from the edge cache or the remote cloud. In the last case, the CI is higher since the UE faces competition for the bottleneck link to access the Internet. We then define a new QoE metric that accounts for the effect of the UE capabilities, the CI, and over/underestimations of the network conditions on the user quality of experience.

Table 1 summarizes the main characteristics of the aforementioned works. As illustrated in the table, our work is the first to approach the joint problem of caching at the network edge and recommendation for on-demand video streaming services taking into consideration context information to improve the user QoE.

TABLE 1  
Comparison of our work with relevant literature on the topic

Paper	Video streaming	Cache at the network edge	Recommendation	QoE-oriented	Context information
[3]	✓	✓	✓	-	-
[14]	✓	-	✓	-	-
[15]	✓	-	✓	-	-
[16]	✓	✓	✓	-	-
[17]	-	-	✓	-	-
[18]	✓	-	-	✓	-
[19]	✓	-	-	✓	-
Our work	✓	✓	✓	✓	✓

## 3 SYSTEM MODEL

We consider a content provisioning scenario as illustrated in Fig. 1. A content provider (e.g., Netflix) offers an on-demand video streaming service within a mobile network infrastructure owned and controlled by a mobile network provider. The video streaming service uses a catalog of video items, where each item is a video content (e.g., The Godfather, The Godfather: Part II, Schindler’s List, Pulp Fiction, etc.) encoded in a certain representation (e.g., medium, large, hd720, etc.). The video streaming service also provides a recommendation system to its users. The mobile network infrastructure comprises a MEC system. In such infrastructure, multiple BSs offer connectivity to a set of mobile users that consume the video streaming service through mobile devices (UEs) connected to the mobile network. MEC servers deployed at the CN host a cache service. Due to the

limited storage capacity, at a given instant of time the cache stores a subset of the catalog, while servers in remote data centers host copies of the entire catalog.

Let us denote by  $\mathcal{U}$  the set of mobile users, by  $\mathcal{V}$  the set of video content, and by  $\mathcal{R}$  the set of all possible video representations. Each video representation  $r \in \mathcal{R}$  is characterized by a bitrate  $Btr(r)$  and a resolution  $Res(r)$ . Let us denote by  $\mathcal{I}$  the catalog of video items. Each video item  $i_{v,r} \in \mathcal{I}$  represents a video content  $v \in \mathcal{V}$  encoded into a representation  $r \in \mathcal{R}$ , so that  $i_{v,r}$  has an encoding bitrate  $Btr(r)$ , a corresponding video resolution  $Res(r)$ , and a size  $Size(i_{v,r})$ . Let us denote by  $C$  the total storage capacity of the cache service, measured in normalized file size units. Table 2 summarizes the notation used in this work.

Next, we describe how we capture user preferences on content, computing and network resources, and the impact of recommendations on user choices in our model.

TABLE 2  
Summary of the formal notation used in this work

Notation	Description
$C$	Total cache storage capacity
$N$	Number of recommendations per user
$\mathcal{U}$	Set of users
$\mathcal{V}$	Set of video content
$\mathcal{R}$	Set of representations (bitrate, resolution)
$\mathcal{I}$	Set of all video items
$\mathcal{T}$	Set of video thematic categories
$\mathcal{Q}$	Set of possible CQI values
$i_{v,r}$	a video item representing a video content $v$ encoded in representation $r$
$Size(i_{v,r})$	Size of the catalog item $i_{v,r}$
$Btr(r)$	Encoding bitrate of video representation $r$
$Res(r)$	Video resolution of video representation $r$
$ResUE(u)$	Maximum resolution supported by the UE of user $u$
$CQI(u)$	Channel Quality Indicator of the UE link of user $u$
$BtrAvl(q)$	Bitrate associated with CQI value $q$
$ResCQI(q)$	Video representation suitable to the CQI value $q$
$\mathbf{f}^v$	Thematic categories vector of the video content $v$
$\mathbf{f}^u$	Thematic preferences vector of user $u$
$Sim(u, v)$	Similarity between preference of user $u$ and video $v$
$P_u^{Cont}$	Video content preference distribution of user $u$
$P_u^{Res}$	Video representation preference distribution of user $u$
$P_u^{Pref}$	Preference distribution of user $u$
$P_u^{Rec}$	Probability distribution due to recommendation
$P_u^{Req}$	Content item request probability distribution of user $u$ (recommended items)
$P_u^{Req\sim}$	Content item request probability distribution of user $u$ (non-recommended items)
$Best(u)$	The most suitable video representation to user $u$
$Sat(u, r)$	Satisfaction of user $u$ with a video representation $r$
$w_u$	Weight that user $u$ gives for a recommendation
$W_u$	Recommendation window of user $u$
$K_u$	Size of recommendation window $W_u$
$Tol(u)$	Distortion tolerance accepted by user $u$

### 3.1 Users and Computing and Network Resources

We assume that each user  $u \in \mathcal{U}$  consumes the on-demand video streaming service through a mobile device (UE) that is connected to a BS. Each UE presents different capabilities (e.g., processor type, graphic card, memory, and screen size) and, consequently, may support a different maximum resolution for video playback, denoted by  $ResUE(u)$ .

At any point in time, each user  $u \in \mathcal{U}$  receives a video streaming through her wireless link. The quality of the UE wireless link may vary over time due to several factors,

including signal impairments. We consider that the quality of the wireless link can be periodically obtained by the video streaming service through the MEC Radio Network Information Service (RNIS) [6]. This quality is reported as the Channel Quality Indicator (CQI), one of the key parameters in the Channel State Information (CSI) [20]. The base station uses the CQI parameter to determine the appropriate Modulation and Coding Scheme (MCS) between itself and the UE. The CQI also influences the transport block size (TBS), which is the size of a data unit, i.e., a transport block from the MAC layer given to the PHY layer. With the MCS and TBS information, the available bandwidth (i.e., the bitrate) of the UE wireless link can be determined.

Let  $\mathcal{Q}$  be the set of all possible CQI values. Each  $q \in \mathcal{Q}$  is associated with a given pair MCS and TBS. Thus, it is possible to determine the available bandwidth of the UE wireless link for each  $q \in \mathcal{Q}$  [21]. We represent such average available bandwidth by  $BtrAvl(q)$ . Also, we denote by  $CQI(u) \in \mathcal{Q}$  the predicted long-term average quality of the UE wireless link of user  $u$  by the time of the recommendation.

### 3.2 Content Preferences

User content preferences are calculated based on the thematic categories of videos that the user has already watched. We assume that a video content is categorized by one or more thematic categories (e.g., Animation, Adventure, Romance, etc.). Formally, let  $\mathcal{T}$  be the set of all possible thematic categories. Each content  $v \in \mathcal{V}$  is associated with a feature vector  $\mathbf{f}^v$  whose  $j$ -th element  $\mathbf{f}^v(j)$ ,  $j \in \mathcal{T}$ , represents the adherence level of category  $j$  to content  $v$ . This adherence level is normalized and assumes values in the range  $[0,1]$ , so that  $\sum_{j \in \mathcal{T}} \mathbf{f}^v(j) = 1, \forall v \in \mathcal{V}$ . More specifically, if a content  $v \in \mathcal{V}$  is classified into  $t$  categories, then the positions corresponding to such categories in the feature vector  $\mathbf{f}^v$  are set to  $1/t$  and the remaining ones to zero.

Each user  $u \in \mathcal{U}$  is also associated with a feature vector  $\mathbf{f}^u$ , where each element  $\mathbf{f}^u(j)$ ,  $j \in \mathcal{T}$ , represents the interest of user  $u$  in thematic category  $j$ . This interest is estimated based on the video content watched and evaluated by the user, denote by  $\mathcal{A}_u \subseteq \mathcal{V}$ . Each element  $\mathbf{f}^u(j)$ ,  $j \in \mathcal{T}$ , of user feature vector  $\mathbf{f}^u$  is estimated as:

$$\mathbf{f}^u(j) = \frac{\sum_{v \in \mathcal{A}_u} \mathbf{f}^v(j)}{\sum_{j \in \mathcal{T}} \sum_{v \in \mathcal{A}_u} \mathbf{f}^v(j)}, \quad (1)$$

and normalized so that  $\sum_{j \in \mathcal{T}} \mathbf{f}^u(j) = 1, \forall u \in \mathcal{U}$ .

Working example 1: Assume a content set  $\mathcal{V} = \{v_1, \dots, v_5\}$  and a category set  $\mathcal{T} = \{1, \dots, 5\}$ , where 1 = "Action", 2 = "Comedy", 3 = "Drama", 4 = "Adventure", and 5 = "Fiction". Suppose that each content  $v \in \mathcal{V}$  is categorized as shown in Table 3 (i.e., each line in the table represents a vector  $\mathbf{f}^v$ ). Assuming that  $\mathcal{A}_u = \{v_4, v_5\}$ , i.e., user  $u$  already watched and evaluated content  $v_4$  and  $v_5$ , these content are used in Equation (1) to compute  $\mathbf{f}^u$ , i.e., the importance of each category for user  $u$ , resulting in:  $\mathbf{f}^u = (\text{"Action"} = 0, \text{"Comedy"} = 0.4170, \text{"Drama"} = 0.4170, \text{"Adventure"} = 0.1658, \text{"Fiction"} = 0)$ .

Given the feature vectors  $\mathbf{f}^v$  and  $\mathbf{f}^u$ , we can estimate the interest of user  $u$  in the video content  $v$ , denoted by

TABLE 3  
Examples of feature vectors ( $\mathbf{f}^v$ ) for 5 videos ( $v_1, \dots, v_5$ )

Content	Action	Comedy	Drama	Adventure	Fiction
$v_1$	0.2	0.2	0.2	0.2	0.2
$v_2$	0	0.33	0	0.33	0.33
$v_3$	0.5	0	0	0	0.5
$v_4$	0	0.33	0.33	0.33	0
$v_5$	0	0.5	0.5	0	0

$Sim(u, v)$ , using the cosine similarity [3], i.e.,

$$Sim(u, v) = \begin{cases} 0, & u \text{ watched } v \\ \frac{\sum_{j \in \mathcal{T}} \mathbf{f}^u(j) \cdot \mathbf{f}^v(j)}{\sqrt{\sum_{j \in \mathcal{T}} \mathbf{f}^u(j)^2} \cdot \sqrt{\sum_{j \in \mathcal{T}} \mathbf{f}^v(j)^2}}, & \text{otherwise.} \end{cases} \quad (2)$$

Thus, for each user  $u \in \mathcal{U}$  and each video content  $v \in \mathcal{V}$  we can compute the probability of user  $u$  being interested in the video content  $v$  as:

$$P_u^{Cont}(v) = \frac{Sim(u, v)}{\sum_{z \in \mathcal{V}} Sim(u, z)}. \quad (3)$$

Equation (3) determines the probability of  $u$  being interested in content  $v$  based on  $u$ 's interest in  $v$  relative to the sum of  $u$ 's interest in each content available in the content set. This distribution is normalized so that  $\sum_{v \in \mathcal{V}} P_u^{Cont}(v) = 1$ . Indeed,  $P_u^{Cont}(v), \forall v \in \mathcal{V}$  represents the content preference distribution of user  $u$  over the content set  $\mathcal{V}$ . We denote this distribution by  $P_u^{Cont}$ .

Working example 2: Having  $\mathbf{f}^u$  and  $\mathbf{f}^v, \forall v \in \mathcal{V}$ , we compute the similarities between user  $u$  and  $v$  (Equation (2)),  $\forall v \in \mathcal{V}$ , and the content preference distribution of this user according to Equation (3). Table 4 shows the results in which the most interesting content for user  $u$  is  $v_1$  with  $P_u^{Cont}(v_1) = 0.4154$ . Content  $v_4$  and  $v_5$  have  $P_u^{Cont}(v) = 0$ , since such content has been already seen by the user.

TABLE 4

Similarities and content preferences examples for user  $u$  and 5 videos

Video	$Sim(u, v)$	$P_u^{Cont}(v)$
$v_1$	0.7298	0.4154
$v_2$	0.5653	0.3218
$v_3$	0.4616	0.2627
$v_4$	0	0
$v_5$	0	0

The demand for video content is time-varying. Usually, it grows for some finite time after the video content becomes available in the catalog, and then it gradually fades out [22], [23]. Similar to [3], we assume that users' content request patterns change slowly over time, usually in the order of a few hours within a day. Thus, content demand predictions, recommendations, and caching decisions are made at this time scale. Another issue related to recommendation system is the cold start problem, i.e., the lack of information on new users [24]. The literature presents several approaches to this problem, from the simple ones (e.g., recommending the most popular cached content) to the sophisticated ones (e.g., employing collaborative systems enriched with external information, such as users' social networks). In this work, we consider only users who have watched and rated a certain number of videos from the content provider.

### 3.3 Representation Preferences

In addition to content interest, the quality in which a video is delivered affects the user experience positively

or negatively. Thus, videos with resolutions that the user mobile device can not reproduce or videos encoded with representations that require more bandwidth than the one available in the UE wireless link should not be among her preferences, even with interesting content.

Recall that  $CQI(u) \in \mathcal{Q}$  denotes the predicted long-term average quality of the UE wireless link of user  $u$  by the time to issue a recommendation. The highest video resolution that can be transmitted under  $CQI(u)$  is given by:

$$ResCQI(CQI(u)) = \max\{Res(r) \mid r \in \mathcal{R} \text{ and } Btr(r) \leq BtrAvl(CQI(u))\}. \quad (4)$$

The best video resolution for user  $u$  by the time of the recommendation, denoted by  $Best(u)$ , is given by the minimum between the video resolution supported by her device and the one supported by her wireless link, i.e.,  $Best(u) = \min\{ResUE(u), ResCQI(CQI(u))\}$ .

Working example 3: Assume the CQI set  $\mathcal{Q} = \{1, \dots, 15\}$  shown in Table 7. For each  $q \in \mathcal{Q}$ , the associated available bandwidth,  $BtrAvl(q)$ , is shown in the fourth column of the table. Consider a representation set  $\mathcal{R} = \{1, \dots, 6\}$  shown in Table 6. For each  $r \in \mathcal{R}$ , the associated bitrate and resolution ( $Btr(r), Res(r)$ ) is shown in the third column of the table. Assume that user  $u$  has a mobile device with resolution  $ResUE(u) = 720px$  and the predicted average quality of her wireless link by the time of the video recommendation is  $CQI(u) = 10$ . According to Equation (4), the highest video resolution that can be transmitted under  $CQI(u)$  is given by  $ResCQI(10) = \max\{360px, 480px, 720px, 1080px, 1440px\}$ . Thus, the best video resolution for user  $u$  is given by  $Best(u) = \min\{720px, 1440px\}$ , i.e., 720px.

To capture the relevance of a given video representation  $r \in \mathcal{R}$  to user  $u$  by the time of the recommendation, we define a representation relevance factor, denoted by  $Sat(u, r)$ , whose values vary in the range  $[0,1]$  and is given by:

$$Sat(u, r) = \begin{cases} 0, & Res(r) > Best(u) \\ 1, & Res(r) = Best(u) \\ 1 - \frac{(Best(u) - Res(r))}{Best(u)}, & Res(r) < Best(u). \end{cases} \quad (5)$$

The rationale behind Equation (5) is that the most appropriate video representation for user  $u$ , given  $ResUE(u)$  and  $ResCQI(CQI(u))$ , achieves the relevance factor of 1, while the other representations show a progressive reduction in the user interest. Representations that are not appropriate to the user mobile device will score 0.

We can compute the representation relevance factor  $Sat(u, r)$  to each video representation  $r$  available in the representation set  $\mathcal{R}$  and determine a new distribution, denoted by  $P_u^{Res}$ , which describes the preferences of user  $u$  on video representations by the time of the recommendation, i.e.,

$$P_u^{Res}(r) = \frac{Sat(u, r)}{\sum_{s \in \mathcal{R}} Sat(u, s)}. \quad (6)$$

Equation (6) determines the probability of  $u$  being interested in representation  $r$  based on  $u$ 's interest in  $r$  relative to the sum of  $u$ 's interest in each representation available in the representation set. This distribution is normalized so that  $\sum_{r \in \mathcal{R}} P_u^{Res}(r) = 1$ . Thus, we distinguish between video content preferences ( $P_u^{Cont}$ ) and video representation preferences ( $P_u^{Res}$ ). We then define a new distribution, called

user preferences ( $P_u^{Pref}$ ) as follows.

**Definition 3.1.** The user preference distribution, denoted by  $P_u^{Pref}$ , is a combination of the user content preference distribution ( $P_u^{Cont}$ ) and the video representation preferences distribution ( $P_u^{Res}$ ) and captures the joint preferences of user  $u$  on a video content  $v \in \mathcal{V}$  and a video representation  $r \in \mathcal{R}$  by the time of the recommendation, for all items in the catalog, i.e.,

$$P_u^{Pref}(i_{v,r}) = \frac{P_u^{Cont}(v) \cdot P_u^{Res}(r)}{\sum_{i_{z,s} \in \mathcal{I}} P_u^{Cont}(z) \cdot P_u^{Res}(s)}, \quad (7)$$

where  $z \in \mathcal{V}$  and  $s \in \mathcal{R}$  are, respectively, the content and representation associated with an item  $i_{z,s} \in \mathcal{I}$ . This distribution is normalized so that  $\sum_{i_{v,r} \in \mathcal{I}} P_u^{Pref}(i_{v,r}) = 1$ . Equation (7) comprises the preferences of user  $u$  in terms of video content and video representation. The latter, on the other hand, depends on the user mobile device capabilities and the UE wireless link condition by the time of the recommendation. Thus, the higher the value of  $P_u^{Pref}(i_{v,r})$ , the better the user experience.

Working example 4: Assuming  $ResUE(u) = 720\text{px}$  and  $CQI(u) = 10$ , according to Equation (5), the representation relevance factor of user  $u$  for each  $r \in \mathcal{R}$  is:  $Sat(u, 1) = 0.33$ ,  $Sat(u, 2) = 0.67$ ,  $Sat(u, 3) = 1$ ,  $Sat(u, 4) = 0$ ,  $Sat(u, 5) = 0$  and  $Sat(u, 6) = 0$ . Suppose a catalog set  $\mathcal{I}$  composed of the items illustrated in the first column of Table 5. The content ( $P_u^{Cont}(v)$ ), representation ( $P_u^{Res}(r)$ ), and preference ( $P_u^{Pref}(i)$ ) distributions for user  $u$  over the catalog  $\mathcal{I}$  according to Equations (3), (6), and (7), respectively, are also shown in Table 5. We observe that the most relevant items for user  $u$  are:  $i_{2,3}$  with  $P_u^{Pref}(i_{2,3}) = 0.2434$  and  $i_{1,2}$  with  $P_u^{Pref}(i_{1,2}) = 0.2105$ .

TABLE 5  
Content, representation, and preference distributions for user  $u$  over the catalog  $\mathcal{I}$  assuming  $ResUE(u) = 720\text{px}$ ,  $CQI(u) = 10$

$i_{v,r}$	$Sim(u, v)$	$P_u^{Cont}(v)$	$Sat(u, r)$	$P_u^{Res}(r)$	$P_u^{Pref}(i_{v,r})$
$i_{1,2}$	0.7299	0.4154	0.67	0.335	0.2105
$i_{1,1}$	0.7299	0.4154	0.33	0.165	0.1037
$i_{2,4}$	0.5654	0.3218	0	0	0
$i_{2,3}$	0.5654	0.3218	1	0.5	0.2434
$i_{2,2}$	0.5654	0.3218	0.67	0.335	0.1631
$i_{2,1}$	0.5654	0.3218	0.33	0.165	0.0803
$i_{3,2}$	0.4616	0.2628	0.67	0.335	0.1332
$i_{3,1}$	0.4616	0.2628	0.33	0.165	0.0656
$i_{4,1}$	0	0	0.33	0.165	0
$i_{5,1}$	0	0	0.33	0.165	0

### 3.4 Impact of Recommendations

The literature shows that recommendation systems have a high impact on user choices. In general, recommendation systems increase requests for recommended items while proportionally decrease the demand for other items in the catalog [25]. There is also strong evidence that the number of recommended items and their respective positions on the recommendation list influence user choices [14], specially for users that access content through mobile devices, where it is less convenient to scroll down the whole recommendation list. To capture the impact of recommendations on user choices, we assume a new probability distribution, denoted by  $P_u^{Req}$ , over the set of video content ( $\mathcal{V}$ ). Formally, let

$\mathcal{V}_u \subseteq \mathcal{V}$  be the set of content recommended to user  $u$ , where  $\mathcal{V}_u$  is composed of the top  $N$  ( $N = |\mathcal{V}_u|$ ) content in  $u$ 's content preference distribution  $P_u^{Cont}$ . Similar to [3], we assume that  $P_u^{Req}(v)$  is a non-increasing function of the content position in the recommendation list ( $pos_u(v) \in [1, N]$ ) and the number of recommended content ( $N$ ). In Section 5.2, we consider a specific instance of  $P_u^{Req}$  that satisfies this assumption. Hence, recommendation systems modulate the original user video preferences  $P_u^{Pref}$  to yield the ultimate video request distribution of user  $u$ , defined as follows.

**Definition 3.2.** Let  $w_u$  be the weight that expresses the importance user  $u$  gives to recommendations. The ultimate video request distribution of user  $u$  over the catalog  $\mathcal{I}$ , denoted by  $P_u^{Req}$ , is a function of the user preference distribution by the time of the recommendation ( $P_u^{Pref}$ ) and the probability distribution due to recommendations ( $P_u^{Res}$ ) and is given by

$$P_u^{Req}(i_{v,r}) = w_u \cdot P_u^{Res}(v) + (1 - w_u) \cdot P_u^{Pref}(i_{v,r}), \quad (8)$$

for the videos that are recommended to user  $u$ , i.e.,  $\forall i_{v,r} \in \mathcal{I} \mid v \in \mathcal{V}_u$ , and by

$$P_u^{Req\sim}(i_{v,r}) = (1 - w_u) \cdot P_u^{Pref}(i_{v,r}), \quad (9)$$

for the videos that are not recommended to user  $u$ , i.e.,  $\forall i_{v,r} \in \mathcal{I} \mid v \notin \mathcal{V}_u$ .

### 3.5 Recommendation Systems as Traffic Engineering

The authors in [3] use recommendation systems to modulate user preferences and, at the same time, optimize content cache policies. To this end, instead of issuing recommendations for the top  $N$  items on the user preference distribution  $P_u^{Pref}$ , as usually expected for a recommendation system, the system selects  $N$  items among the ones residing within a recommendation window  $W_u$ . This window is defined by the top  $K_u$  items of  $P_u^{Pref}$ , where  $K_u > N$ . Indeed,  $K_u$  is defined according to a preference distortion tolerance  $Tol(u) \in [0, 1]$ . This parameter quantifies how much the recommendations may distort the original user preferences. Higher  $Tol(u)$  values allow for more flexibility in selecting items to recommend to users and so favor caching efficiency.

Given a list  $W_u$ ,  $W_u \subseteq \mathcal{I}$ , with  $K_u$  items, the last  $N$  items will be recommended in the worst case, i.e., those ranked in positions  $K_u - N + 1, K_u - N + 2, \dots, K_u$ . Denoting by  $i$  an item in  $W_u$  and by  $pos_u(i)$  the position of  $i$  in  $W_u$ , the worst case distortion is defined as:

$$\Delta_u(K_u, N) = 1 - \frac{\sum_{i: pos_u(i) \in [K_u - N + 1, K_u]} P_u^{Pref}(i)}{\sum_{i: pos_u(i) \in [1, N]} P_u^{Pref}(i)}. \quad (10)$$

The cardinality  $K_u$  of the expanded recommendation window  $W_u$  is given by:

$$K_u = \max\{k \mid \Delta_u(K_u, N) \leq Tol(u)\}. \quad (11)$$

In this work, we use the same approach proposed in [3] to modulate user preferences and to optimize cache policies. However, the definition of preferences in [3] does not account for video representations, and each item  $i \in \mathcal{I}$  is only described by its content (i.e.,  $i = v$ ). As a consequence,  $P_u^{Pref} = P_u^{Cont}$ . Different from [3], the definition of preferences in our work takes into account video content ( $v$ ) and representation ( $r$ ). Thus, each item  $i \in \mathcal{I}$  is a video content encoded into a certain representation (i.e.,  $i = i_{v,r}$ )

and user preferences are defined in terms of both content and representation, as stated in Equation (7).

#### 4 CACHE POLICY AND RESOURCE-AWARE RECOMMENDATION SYSTEM

In this section, we define a new joint cache and recommendation problem, called *Resource-Aware Video Recommendation* (RAViR), which takes into consideration content preferences and information on the availability of computing and network resources by the time of the video recommendation.

Consider a cache service running in MEC hosts at the CN with total storage capacity  $C$ ; the catalog  $\mathcal{I}$  composed of video content, their representations, and their lengths  $Size(i_{v,r})$ ,  $i_{v,r} \in \mathcal{I}$ ; the set  $\mathcal{U}$  of users with their distributions of preferences ( $P_u^{Pref}$ ,  $u \in \mathcal{U}$ ) and request ( $P_u^{Req}$ ,  $u \in \mathcal{U}$ ) given by Equations (7) and (8) and (9), respectively; and user recommendation windows  $W_u \subseteq \mathcal{I}$  with  $K_u$  videos, where  $K_u$  is given by Equation (11) and  $u \in \mathcal{U}$ . The goal of RAViR is to recommend, for every user  $u \in \mathcal{U}$ ,  $N$  videos from the recommendation window  $W_u$ , so that the recommended videos maximize the cache hit ratio and the user experience.

Formally, let  $\{y_{i_{v,r}}\}$ ,  $i_{v,r} \in \mathcal{I}$ , be a set of binary decision variables, so that  $y_{i_{v,r}} = 1$  if video  $i_{v,r}$  is in cache and  $y_{i_{v,r}} = 0$ , otherwise. Let  $\{x_{u,i_{v,r}}\}$ ,  $u \in \mathcal{U}$ ,  $i_{v,r} \in \mathcal{I}$ , be another set of binary decision variables, so that  $x_{u,i_{v,r}} = 1$  if  $i_{v,r}$  is recommended to user  $u$  and  $x_{u,i_{v,r}} = 0$ , otherwise. We formulate the RAViR problem as follows:

$$\max_{y,x} \sum_{u \in \mathcal{U}} \sum_{i_{v,r} \in W_u} y_{i_{v,r}} (x_{u,i_{v,r}} \cdot P_u^{Req}(i_{v,r}) + (1 - x_{u,i_{v,r}}) \cdot P_u^{Req\sim}(i_{v,r})) \quad (12)$$

subject to:

$$\sum_{i_{v,r} \in \mathcal{I}} y_{i_{v,r}} \cdot Size(i_{v,r}) \leq C \quad (13)$$

$$\sum_{i_{v,r} \in W_u} x_{u,i_{v,r}} = N, \forall u \in \mathcal{U} \quad (14)$$

$$P_u^{Res}(r) > 0, \forall u \in \mathcal{U}, \forall i_{v,r} \in W_u. \quad (15)$$

Equation (12) maximizes the cache hit ratio and the user experience using each user preference list  $W_u$ . This list comprises preferences on content (explicitly) and resolution (implicitly). Equation (13) reflects the cache storage capacity constraint, while Equation (14) ensures that exactly  $N$  videos are recommended to every user. Finally, Equation (15) ensures that the recommended video does not exceed the computing and network resources to every user.

#### Problem Complexity

As the authors of [3], we identify the complexity of our formulation from the polynomial reduction of the 0-1 Knapsack Problem (KSP), a NP-complete problem [26], where each item  $i_{v,r}$  in the catalog  $\mathcal{I}$  has a size  $Size(i_{v,r})$  and a value  $\sum_{u \in \mathcal{U}} (x_{u,i_{v,r}} \cdot P_u^{Req}(i_{v,r}) + (1 - x_{u,i_{v,r}}) \cdot P_u^{Req\sim}(i_{v,r}))$ , and have to be packed within a cache space  $C$ . In this way, a valid solution for a 0-1 KSP instance corresponds to a valid solution for a RAViR instance. Furthermore, if there is no solution in 0-1 KSP, there is also no solution in RAViR. Moreover, assuming a RAViR decision problem as a decision version of a 0-1 KSP, and that there is an oracle

that solves it and returns the recommendation lists for all users and the cache placement, we can directly solve this decision version of the 0-1 KSP in a polynomial time. Thus, the RAViR problem, represented by Equations (12)-(15), is a NP-complete problem. Despite the problem being NP-complete, we are able to solve instances of interest in a reasonable time. To illustrate, while employing the dataset used in [3] as input for our evaluation, RAViR consumes approximately 24 minutes in a computer with a quad-core Intel Core i7 @ 2.7 GHz and 16 GB RAM. Our algorithm is implemented using Python 2.7.12, docplex 2.8.125, and IBM CPLEX 12.8.0 as the solver.

## 5 EXPERIMENTAL EVALUATION

This section evaluates the performance of RAViR using a video catalog derived from real-world data. In the following, we first describe the methodology we use to obtain the video catalog (Section 5.1). Then, we i) detail how we obtain information on computing and network resources related to the user mobile devices (Section 5.2), ii) present the baseline algorithms used in the evaluation (Section 5.3), and iii) introduce the metrics used to assess the performance of the evaluated methods (Section 5.4). Finally, we discuss the obtained results (Sections 5.5 and 5.6).

### 5.1 Dataset and Pre-processing

In order to create the video catalog, we use the MovieLens project dataset [11] to obtain video content and user content preferences. The MovieLens dataset is a collection of 5-star movie ratings collected from an online movie recommendation service called MovieLens. This dataset has been used in related work on caching and recommendations [3], [27] and consists of approximately 100,000 user ratings (ranging from 0 to 5) applied to a content catalog of 9,742 movies ( $|\mathcal{V}| = 9,742$ ), characterized by 20 thematic categories ( $|\mathcal{T}| = 20$ ), and a community of 610 users ( $|\mathcal{U}| = 610$ ). All users employed in the evaluation have rated at least 20 movies from the dataset.

Since the MovieLens dataset does not provide different video representations for content, we use video qualities supported by YouTube to derive a set of video representations. We consider 6 video resolutions, and their corresponding YouTube recommended bitrates [21], shown in Table 6, to compose the set of video representations ( $|\mathcal{R}| = 6$ ).

TABLE 6

Video representations and corresponding labels used in our evaluation

Type $r$	Label	Video Representation $(Btr(r), Res(r))$
1	medium	(1.5Mbps, 360px)
2	large	(4Mbps, 480px)
3	hd720	(7.5Mbps, 720px)
4	hd1080	(12Mbps, 1080px)
5	hd1440	(24Mbps, 1440px)
6	hd2160	(53Mbps, 2160px)

We then combine this video representation set ( $\mathcal{R}$ ) and the MovieLens dataset ( $\mathcal{V}$ ) to generate the final video catalog  $\mathcal{I}$ . For each video content  $v \in \mathcal{V}$ , we generate a number  $n_v$  that indicates the amount of representations of  $v$  available in the catalog  $\mathcal{I}$ , with  $n_v$  sampled from a Uniform distribution

$U(1, 6)$ . Given  $v$  and  $n_v$ , we insert in the catalog  $\mathcal{I}$  all items  $i_{v,r}$  with content  $v$  encoded in representation  $r$ , so that  $r \in \{1, \dots, n_v\}$ . This approach ensures that every video content  $v \in \mathcal{V}$  is available in a subset of representations with increasing video qualities, starting from the lowest one. The size of the video ( $Size(i_{v,r}), i_{v,r} \in \mathcal{I}$ ) is a function of its duration and representation, employing a normalized scale compatible with the cache storage. After combining the MovieLens dataset ( $\mathcal{V}$ ) and the set of representations ( $\mathcal{R}$ ), we have a video catalog of near 35,000 items ( $|\mathcal{I}| = 35,000$ ).

We also use the MovieLens dataset ( $\mathcal{V}$ ) to compute the feature vectors  $\mathbf{f}^v$  and  $\mathbf{f}^u$  and then infer the user content preference  $P_u^{Cont}$ . To this end, we use the same approach as in [3] and assume that if the user rated and watched a content, the user is interested in the thematic categories of such content, regardless of the actual user rating. This approach is reasonable since the actual user rating depends on several factors. For instance, a low rating can be associated with a bad script. Once the feature vectors  $\mathbf{f}^v$  and  $\mathbf{f}^u$  are computed for every  $v \in \mathcal{V}$  and for every  $u \in \mathcal{U}$ , respectively, we use Equations (2) and (3) to estimate  $P_u^{Cont}$  for every  $u \in \mathcal{U}$ .

### 5.2 Computing and Network Resources and Ultimate User Request

As part of the input to our model, we need to provide computing capabilities related to the user mobile device and the prediction of the average quality of the UE wireless link by the time of the recommendation. To represent the UE computing capabilities, we assign the maximum supported resolution ( $ResUE(u), u \in \mathcal{U}$ ) by taking a random value from a Uniform distribution  $U(1, 6)$ . This ensures that  $ResUE(u) = Res(r)$  for some video representation  $r$  described in Table 6. Similar to the previous random choices employed, this simple approach has the sole purpose of exercise the model without bias. Any statistical knowledge about UE computing capabilities or video representations could influence the specific values observed in the evaluation, but not the general trends.

For the quality of the UE wireless link ( $CQI(u) \in \mathcal{Q}, u \in \mathcal{U}$ ), values of signal strength-related metrics (e.g., the Arbitrary Strength Unit - ASU) can be grouped into CQI values, e.g.,  $|\mathcal{Q}| = 15$ , as illustrated in Table 7 (column 1) [21]. Given that each CQI value  $q \in \mathcal{Q}$  is associated with an MCS and a TBS, there is a mapping from CQI values to bitrates, also illustrated in Table 7 (column 4). Combining the information presented in Tables 6 and 7, it is possible to determine the best video representation for a certain CQI value. This information is shown on the right side of Table 7 (column 5). Again, in order to avoid bias, we obtain the predicted average quality of the UE wireless link ( $CQI(u)$ ) by sampling from a Uniform distribution  $U(1, 15)$  (i.e., the interval presented in column 1 of Table 7).

Having  $ResUE(u)$  and  $CQI(u), u \in \mathcal{U}$ , we can compute the best representation for user  $u$  by the time of the recommendation. Also, using Equations (5), (6), and (7), we can estimate  $P_u^{Pref}, \forall u \in \mathcal{U}$ . Given  $P_u^{Pref}$  and  $Tol(u)$ , we calculate the cardinality ( $K_u$ ) of the  $W_u$  using Equations (10) and (11), and the first  $K_u$  elements of the distribution  $P_u^{Pref}$  are inserted in the  $W_u$ . Similar to [3], we assume that recommendations provide all  $N$  content in  $\mathcal{V}_u$  with an equal boost, i.e.,  $P_u^{Rec}(v) = 1/N, \forall v \in \mathcal{V}_u$ . In our evaluation, we

test with  $N = 3$  and  $N = 5$ . Given  $P_u^{Rec}$  and  $P_u^{Pref}, \forall u \in \mathcal{U}$ , we compute the ultimate video request distribution ( $P_u^{Req}$  and  $P_u^{Req^*}, \forall u \in \mathcal{U}$ ) using Equations (8) and (9). Aligned with [3], we sample the user recommendation weight ( $w_u, \forall u \in \mathcal{U}$ ), from a Uniform distribution  $U(0.5, 0.7)$ .

TABLE 7  
Mapping between CQI values, bitrates, and video representations

CQI $q$	MCS	TBS	Bitrate $BtrAvl(q)$	Video Representation $ResCqi(q)$
Value	Index	(bits)	(Mbps)	(Mbps,px)
1	0	1384	2.768	(1.5Mbps,360px)
2	0	1384	2.768	(1.5Mbps,360px)
3	2	2216	4.432	(4Mbps,480px)
4	4	3624	7.548	(7.5Mbps,720px)
5	6	5160	10.320	(7.5Mbps,720px)
6	8	6968	13.936	(12Mbps,1080px)
7	11	8760	17.520	(12Mbps,1080px)
8	13	11448	22.896	(12Mbps,1080px)
9	16	15264	30.528	(24Mbps,1440px)
10	18	16416	32.832	(24Mbps,1440px)
11	21	21384	42.768	(24Mbps,1440px)
12	23	25456	50.912	(24Mbps,1440px)
13	25	28336	56.672	(53Mbps,2160px)
14	27	31704	63.408	(53Mbps,2160px)
15	27	31704	63.408	(53Mbps,2160px)

### 5.3 Baseline Methods

We compare RAViR against JCRP. However, user preferences in JCRP are based on video content and do not account for video representation. Thus, to provide a fair comparison between the two methods, we adapt the JCRP model to limit the video representations considered during the generation of the user recommendation window  $W_u$ . This adaptation does not affect the output issued by the method since, for JCRP, videos with different representations, but the same content, have the same value for the user. This approach leads us to three variants of the original method:

- JCRP-L: given a set of videos with the same content, this variant selects the representation with the lowest resolution to be included in  $W_u$ .
- JCRP-M: given a set of videos with the same content, this variant selects the representation with medium resolution to be included in  $W_u$ .
- JCRP-H: given a set of videos with the same content, this variant selects the representation with the highest resolution to be included in  $W_u$ .

### 5.4 Evaluation Metrics

Our evaluation is focused on two metrics: the user QoE and the cache hit ratio (CHR). In the following, we describe these metrics in detail.

The QoE metric estimates the user's satisfaction when watching a video. In our work, this metric is computed for each video recommended to the user. The QoE is high when the recommended video is in cache and its representation matches the computing and network resources available to the UE. More specifically, the following factors affect the user QoE negatively:

- **F1**: The recommended video is not in cache. In this case, the video will be delivered by the cloud server,

and the UE has to face competition for the bottleneck link to access the Internet.

- **F2:** The recommended video requires a higher bitrate than that available on the UE wireless link. In this case, the video reproduction is subjected to delays and stalls.
- **F3:** The recommended video requires a higher resolution than the one supported by the user mobile device. In this case, the UE is not able to reproduce the video.
- **F4:** The recommended video requires a lower resolution than that supported by the available resources (UE computing capabilities and wireless link). In this case, the available resources are not being used properly and the user is receiving a video with lower quality than her device is able to present.

In [19], the authors propose a correlation between the user QoE and the Congestion Index ( $CI$ ), a metric that represents the ability of the network to successfully deliver a video stream based on the maximum available bandwidth on the path from the server to the user. This correlation is shown in Equation (16):

$$QoE = \frac{5.082}{\sqrt{CI}} - 0.891, \quad CI > 0, \quad (16)$$

where  $CI$  is the ratio of the maximum required bandwidth for the stream divided by the available network bandwidth. The authors in [19] obtain this equation using a regression technique over subjective user assessments.

In our work, the recommended video can be delivered by the MEC hosts or by the cloud servers, depending on whether the video is in the cache or not. To capture this situation, we define an Internet Congestion Index metric, denoted by  $ICI$ , that represents the current state of the network path between the MEC hosts and the cloud servers. If the video is in the cache, no access to the cloud servers is needed, and  $ICI = 0$ . Otherwise,  $ICI > 0$  and its value will depend on the utilization of the links connecting the two storage places (i.e., MEC and cloud). We then propose a new  $CI$  ratio to account for the  $ICI$  ratio as follows. Given an item  $i_{v,r} \in \mathcal{I}$  recommended to user  $u \in \mathcal{U}$ , we compute the  $CI$  for such recommendation as:

$$CI(u, r) = \frac{Btr(r)}{BtrAvl(CQI(u))} * (1 + ICI), \quad (17)$$

where  $Btr(r)$  and  $BtrAvl(CQI(u))$  represent, respectively, the maximum required bitrate for the stream and the bitrate available in the UE wireless link.

Equations (16) and (17) work well to capture the user QoE concerning video transmission, e.g., the situations described in **F1** and **F2**. However, it fails to capture the QoE in other situations. For instance, for situation **F4**, these equations return a high user QoE, when it is expected otherwise. To capture the QoE of user  $u \in \mathcal{U}$  when receiving a recommended video  $i_{v,r} \in \mathcal{I}$  and taking into account all situations described from **F1** to **F4**, we propose a new QoE metric, based on Equation (16), as follows:

$$QoE(u, i_{v,r}) = \begin{cases} 0, & Res(r) > Best(u) \\ \frac{5.082}{\sqrt{CI(u,r)}} - 0.891, & Res(r) = Best(u) \\ \frac{5.082}{\sqrt{1/CI(u,r)}} - 0.891, & Res(r) < Best(u), \end{cases} \quad (18)$$

where  $CI$  is given by Equation (17). The first part of Equation (18) captures the user QoE in case of **F3** happens; the second part of the equation applies for **F1** and **F2**. Finally, the third part deals with **F4**.

For the cache hit ratio (CHR), we adapt the metric employed in [3], which is based on the probability of cached videos being requested. In [3] and in most literature on video caching, a recommendation is considered a hit if the video content is in the cache. These works either assume that all representations (of the content) are in the cache, or that the highest representation is cached and transcoding techniques will be in charge of transforming such representation to the one expected by the user. Both strategies, however, rapidly consume the available cache storage. Indeed, Netflix has reported achieving the same caching efficiency with 50% less storage by caching videos at a file (representation) level instead of at a title (content) level [28]. Thus, in this work, we assume that cache policies are defined at the granularity of files (representations) and consider a cache hit only if the cached file matches the content and representation expected by the user (according to its computing and network resources). To reflect this situation, we adapt the CHR metric by introducing a binary variable  $z_{u,i_{v,r}}$  that assumes the value 1 only when the cached video matches perfectly with the content and representation expected by the user, and 0 otherwise. Thus, our cache hit ratio is given by:

$$CHR = \frac{\sum_{u \in \mathcal{U}} \sum_{i_{v,r} \in \mathcal{I}'} z_{u,i_{v,r}} \cdot P_u^{Req}(i_{v,r})}{\sum_{u \in \mathcal{U}} \sum_{i_{v,r} \in \mathcal{I}} P_u^{Req}(i_{v,r})}, \quad (19)$$

where  $\mathcal{I}' \subset \mathcal{I}$  is the set of videos stored in the cache.

## 5.5 Evaluating Content, UE Capabilities and Network Conditions

In this section, we evaluate the performance of RAViR and the JCRP variants considering the entire input needed for the model, i.e., video content, user mobile device capabilities, and the quality of the UE communication (including both, UE wireless link and Internet access). We first present the results related to the QoE, considering a fixed  $ICI$  (Section 5.5.1). Then, we present the results related to the  $CHR$  (Section 5.5.2). Finally, we present an analysis on the impact of the  $ICI$  on the QoE (Section 5.5.3).

### 5.5.1 Performance regarding the QoE

The output of RAViR and the JCRP variants is a list with  $N$  recommended videos for each user. To understand how these methods perform in relation to the recommendations and how such recommendations affect the overall QoE perceived by the users, we classify the recommendations into four types:

- **edge\_rec:** recommendation whose video is in the cache and factors **F2**, **F3** and **F4** do not occur;
- **cloud\_rec:** recommendation whose video is not in the cache and factors **F3** and **F4** do not occur;
- **under\_rec:** recommendation whose video resolution is below than what could be reproduced by the UE according to the available resources;
- **over\_rec:** recommendation whose video resolution is higher than what the UE is able to reproduce.

For this evaluation, we consider three scenarios where each user receives a list of  $N$  recommended videos according to an distortion tolerance ( $Tol(u)$ ), namely:

- Scenario 1: with  $N = 3$  and  $Tol(u) = 1\%$ ,  $\forall u \in \mathcal{U}$ ;
- Scenario 2: with  $N = 3$  and  $Tol(u) = 10\%$ ,  $\forall u \in \mathcal{U}$ ;
- Scenario 3: with  $N = 5$  and  $Tol(u) = 1\%$ ,  $\forall u \in \mathcal{U}$ .

The most suitable recommendation corresponds to the video cached at the network edge with the representation proper to the user resources (edge\_rec). The higher the percentage of this type of recommendation, the greater the accuracy of the method. In this evaluation, we present the percentage of edge\_rec, under\_rec, over\_rec, and cloud\_rec recommendations issued by each method. Fig. 2 shows the percentage of recommendations that fall into each recommendation type (left Y-axis), as well as the normalized average QoE (right Y-axis) achieved by each method, considering the three scenarios previously described. In this evaluation,  $ICI = 60\%$ , while the total cache capacity ( $C$ ) varies in each column of graphics, with the values 50, 300, and 700. Each row exhibits the results from one scenario. As expected, by increasing the total cache capacity from 50 to 700, the percentage of recommendations in the cache (edge\_rec) increases, while the percentage of recommendations in the cloud (cloud\_rec) decreases in all methods, for all scenarios.

When a large percentage of recommendations made by a certain method falls into the under\_rec or over\_rec type, the QoE is low. Similarly, as the percentage of recommendations falling into the cloud\_rec type increases, the QoE decreases. However, the (negative) impact of under\_rec and over\_rec recommendations on the QoE is higher than that of cloud\_rec recommendations. This is illustrated in Fig. 2(a), where the percentage of cloud\_rec recommendations issued by RAViR is the highest among all evaluated methods but, despite this fact, the QoE is the highest for this scenario.

Since scenario 2 (Figs. 2(d), 2(e), and 2(f)) has a higher user distortion tolerance in comparison with scenario 1, the methods have more flexibility to accommodate the different user preferences in the cache. As a consequence, more recommendations fall into the edge\_rec type and the QoE increases for all methods. On the other hand, scenario 3 (Figs. 2(g), 2(h) and 2(i)), that has a higher number of recommended videos than scenario 1 but the same user distortion tolerance, becomes more dependent on content provided by the cloud. As a consequence, the QoE is slightly lower for all methods in this scenario.

Overall, we observe that the QoE achieved by RAViR is notably higher than the ones achieved by the JCRP variants in all scenarios and regardless of the total cache capacity. This happens because RAViR takes into account computing and network resources at the moment of its recommendations, avoiding the undesirable recommendation types of under\_rec and over\_rec. The poor performance of all JCRP variants comes from their unawareness of the computing and network resources and the consequent design focused only on the content cache hit. As a consequence, most of the time, JCRP variants recommend videos with inadequate representations (under\_rec or over\_rec). JCRP-L presents the best performance among the JCRP variants since it achieves the highest cache hit ratio and never generates over\_rec. However, JCRP-L is overcome by RAViR even with the

smallest cache size (i.e., 50) which corresponds to its most favorable configuration. The reason is the high level of under\_rec obtained by JCRP-L which represents the dissatisfaction of users who have enough device and network resources to receive a good quality video but receive a poor quality one.

Fig. 3 summarizes the performance of the methods in terms of average user QoE but presenting a larger range of cache capacities (from 50 until 1000). RAViR presents the expected behavior that consists of increasing QoE as a function of the cache capacity. This increase tends to become less relevant as the cache system approaches the optimal condition, i.e., when all videos are cached. Naturally,  $Tol(u) = 10\%$  (Fig. 3(b)) presents better results than  $Tol(u) = 1\%$  (Fig. 3(a)) because the sharing opportunities are higher. The overall behavior of JCRP-L is similar to RAViR, but exhibiting a notably lower performance since only the lowest representation is cached while many UEs can consume content with higher quality. The behavior of JCRP-M and JCRP-H seems inconsistent since the increase in the cache capacity sometimes implies decreasing QoE. However, this is consistent with the representation unawareness of the method, which implies storing content with representations that can not be consumed by certain UEs due to limitations in its computing or wireless link capacities. This evaluation illustrates numerically the potential benefits of RAViR. For example, RAViR shows an increase from 68% to 85% in QoE when compared with JCRP-L in all cache sizes, and more than 100% when compared with the other methods (JCRP-M and JCRP-H).

We also evaluate QoE as a function of the distortion tolerance, considering the smallest cache capacity ( $C = 50$ ) and high Internet congestion ( $ICI = 60\%$ ), as shown in Fig. 4. Since JCRP-L can always keep more videos in the cache than other methods, its performance is not notably affected by the distortion tolerance. On the other hand, JCRP-M, JCRP-H, and RAViR exhibit a more perceivable improvement with the increase of the distortion tolerance since this represents more receptiveness from the users to diverse content. As expected, RAViR presents a more consistent increase since it is aware of the QoE while JCRP variants are not.

### 5.5.2 Performance regarding the CHR

This evaluation is focused on the cache hit ratio metric ( $CHR$ ), which was described at the end of Section 5.4. Similar to the previous section, we consider three and five recommendations per user ( $N = 3$  and  $N = 5$ ) and the distortion tolerance accepted by the users ( $Tol(u)$ ) assumes 1%. The results obtained with a distortion tolerance of 10% are similar and are omitted due to space constraint.

Fig. 5 shows the performance of the methods in terms of the  $CHR$  as a function of the cache capacity, which varies from 50 to 1000. Fig. 5(a) shows the results obtained according to Equation (19), where over\_rec and under\_rec recommendations are not considered as cache hit. There is some similarity between the  $CHR$  and the QoE results shown in Fig. 3(a), although the  $CHR$  is more sensitive to the cache capacity. Again, RAViR has a more consistent behavior than the JCRP variants, with the  $CHR$  increasing as the cache capacity increases. This consistency is also observed when comparing the results with three and five recommendations per user. As expected, the  $CHR$

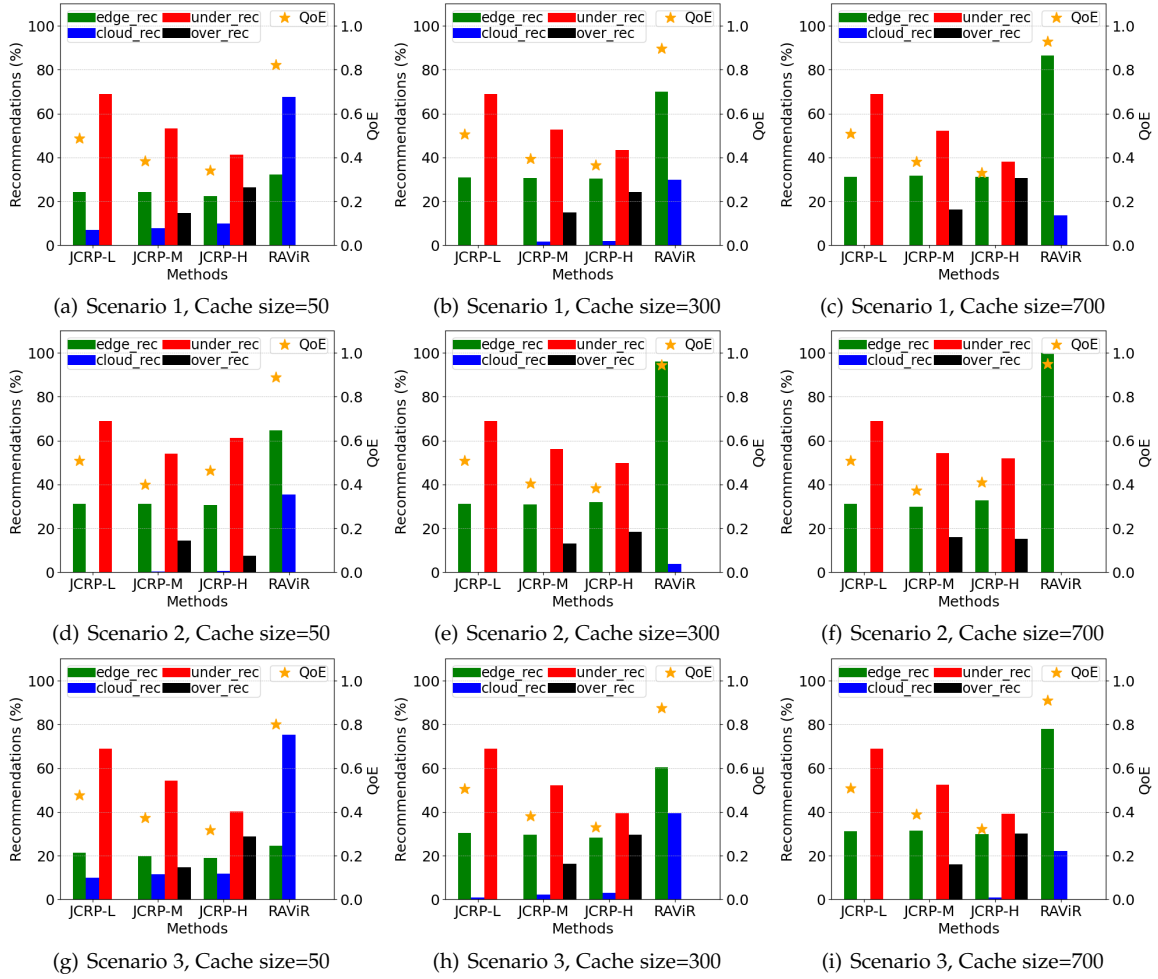


Fig. 2. Percentage of recommendations by type and average user QoE, for each method, in the three considered scenarios. Total cache capacity assumes the values 50, 300, and 700.

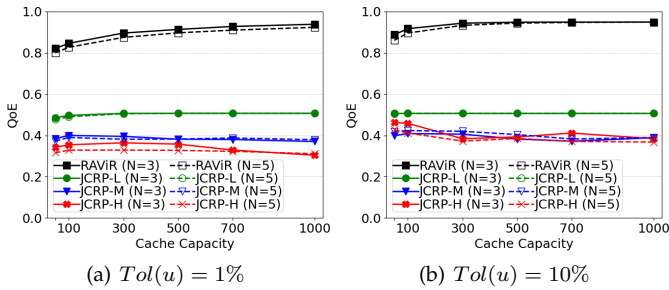


Fig. 3. Impact of the total cache capacity on the average user QoE achieved by RAViR and JCRP variants. In the figures, solid lines represent experiment with 3 recommendations per user ( $N = 3$ ) and dashed lines represents experiment with 5 recommendations per user ( $N = 5$ ).

achieved by RAViR with three recommendations per user is higher than that with five. The poor performance of all JCRP variants was expected due to their unawareness of the computing and network resources. JCRP-L is penalized mostly due to under\_rec recommendations, JCRP-M suffers with under and over\_rec recommendations, while JCRP-H is penalized mostly by over\_rec recommendations. JCRP-L performs slightly better than the other JCRP variants since it can keep more videos in the cache, as it operates with lower video representations. Overall, the effective  $CHR$  achieved by RAViR is at least 14% higher than that of JCRP-L.

Fig. 5(b) shows the same evaluation of Fig. 5(a) but

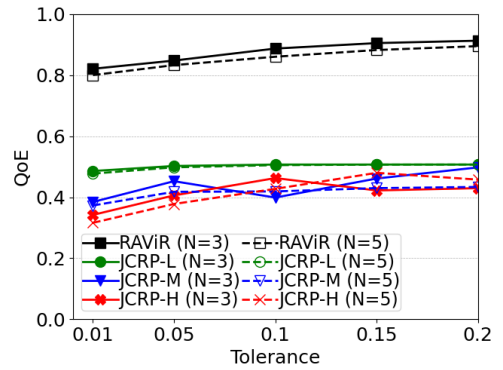


Fig. 4. Impact of preference distortion tolerance on the QoE for RAViR and the JCRP variants. This analysis considers the total cache capacity of 50 ( $C = 50$ ), Internet Congestion Index ( $ICI$ ) of 60%, and 3 and 5 recommendations per user ( $N = 3$  and  $N = 5$ ).

using a more conservative definition for the  $CHR$ , where only over\_rec recommendations are not considered cache hit. In this case, when the total cache capacity is small, the JCRP variants perform better than RAViR. This happens because RAViR pursues a perfect match of content and representation for each user, making sharing more difficult among users. For the JCRP variants, however, as long as a cached video matches with the recommended content and does not lead to over\_rec recommendation, it can be shared

among users. While this more conservative definition of *CHR* limits the sharing opportunities for RAViR, it also shows, by comparing Figs. 5(b) and 3(a), that high *CHR* does not imply in high user QoE. Additionally, this conservative definition does not reflect the way DASH (Dynamic Adaptive Streaming over HTTP) clients, commonly used in video streaming, work in practice. In general, DASH clients seek to obtain chunks in the best video representation supported by the UE given the limitations imposed by network throughput.

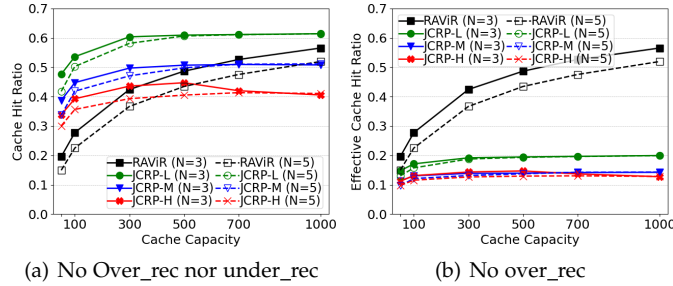


Fig. 5. Performance of RAViR and the JCRP variants concerning the CHR for different total cache capacities. In figure (a) *over\_rec* and *under\_rec* recommendations are not considered as cache hit, and in figure (b) *over\_rec* recommendations are not considered as cache hit. Solid lines represent experiment with 3 recommendations per user ( $N = 3$ ) and dashed lines represent 5 recommendations per user ( $N = 5$ ).

Additionally, we evaluate the effective cache hit ratio as a function of the distortion tolerance, considering the smallest cache capacity ( $C = 50$ ) and high Internet congestion ( $ICI = 60\%$ ), as shown in Fig. 6. All methods present noticeable improvement with the increase of the distortion tolerance, but RAViR explores more efficiently the increase in the receptiveness from the users to diverse content. JCRP variants are negatively affected by the mismatches between the cached content and the representation expected by the users, while RAViR was designed to avoid this issue.

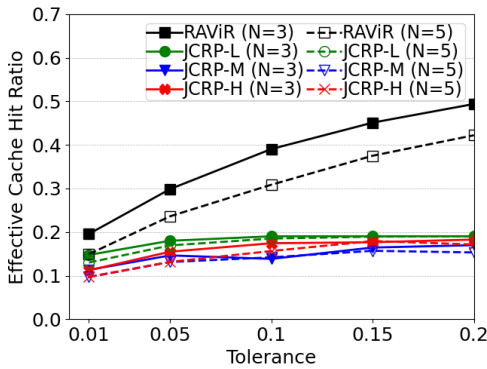


Fig. 6. Impact of preference distortion tolerance on the effective cache hit ratio for RAViR and the JCRP variants. This analysis considers the total cache capacity of 50 ( $C = 50$ ), Internet Congestion Index ( $ICI$ ) of 60%, and 3 and 5 recommendations per user ( $N = 3$  and  $N = 5$ ).

### 5.5.3 Impact of the Internet Congestion Index on QoE

To evaluate the impact of the Internet Congestion Index ( $ICI$ ) on the QoE, we vary this index from 10% to 70%, representing a wide range of network conditions. Naturally, if the recommended video is in the cache, then the UE does not need to access the Internet, i.e.,  $ICI = 0$ . The  $ICI$  has a

major impact on scenarios where the total cache capacity is small. Thus, this evaluation employs a total cache capacity of 50 ( $C = 50$ ). The experiments are carried out considering three and five recommendations per user ( $N = 3$  and  $N = 5$ ), and a distortion tolerance accepted by users of 1% ( $Tol(u) = 1, \forall u \in \mathcal{U}$ ).

Fig. 7 shows the average QoE as a function of the  $ICI$ . On one hand, all JCRP variants are unaware of the network conditions, which makes them less sensitive to the  $ICI$  than RAViR. On the other hand, the performance of RAViR is still notably superior to all JCRP variants even in the worst evaluated case. For example, under  $ICI = 70\%$ , the smallest advantage of RAViR ( $N = 5$ ) over JCRP-L ( $N = 3$ ), the JCRP variant with the best performance, is nearly 62%.

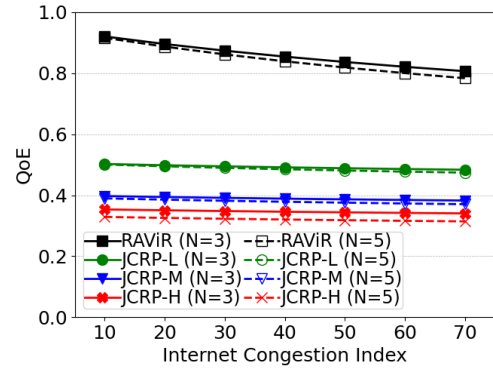


Fig. 7. Impact of the Internet Congestion Index ( $ICI$ ) on the QoE for RAViR and the JCRP variants. This analysis considers user distortion tolerance of 1% ( $Tol(u) = 1$ ), total cache capacity of 50 ( $C = 50$ ), and 3 and 5 recommendations per user ( $N = 3$  and  $N = 5$ ).

## 5.6 Highlighting the Network Conditions

Nowadays, user mobile devices are still very heterogeneous and their capabilities to present video content varies significantly. However, the continuous evolution of these devices, in general, followed by large drops in their prices, may change the context in a near future. Under this assumption, the network conditions become the relevant impact factor in the consumption of the video content. In this section, we evaluate the methods assuming that all UEs have computing capabilities to play video content in any resolution, constrained only by the network conditions.

In this context, we need to update some concepts in our model. Initially, we redefine  $Best(u)$  so that the best video representation for user  $u$  is the one that best meets the requirements of the quality of its UE wireless link, i.e.,  $Best(u) = ResCQI(CQI(u))$ .

Based on the new values of  $Best(u)$ , we recompute our model ( $P_u^{Res}$ ,  $P_u^{Pref}$ , and  $P_u^{Req}$ ) and find the new recommendation window  $W_u$ , for every  $u \in \mathcal{U}$ . We then solve the RAViR problem removing the user mobile device constraint. Note that, in this redefinition of the problem, the QoE is now affected by factors **F1**, **F2**, and **F4**. Additionally, the recommendations are now classified into three types: *edge\_rec*, *cloud\_rec*, or *under\_rec*.

Fig. 8 shows the percentage of recommendations that fall into each recommendation type (left Y-axis) and the normalized average QoE (right Y-axis) achieved by each method, considering Scenario 1 described in Section 5.5.1.

Scenarios 2 and 3 are omitted for the sake of brevity, but they provide the same conclusions. Figs. 8(a), 8(b), and 8(c) show the result for a total cache capacity of 50, 300, and 700, respectively. Comparing the results presented in Fig. 2 (from Section 5.5.1) with the results shown in Fig. 8, there is a large decrease in the average QoE achieved by all methods. The reason is the update in the users' expectations that were originally partially represented by their devices capabilities, and are now unconstrained in terms of device resolution (i.e., all UEs can play the highest video resolution). At the same time, the network conditions are the same as the previous evaluation, meaning an increased demand with the same amount of resources (both network and MEC). Thus, the overall decrease in the average QoE is a consistent result. Since video resolution is not constrained in any UE, the JCRP-H variant becomes the one with the best performance, since now it experiences under\_rec recommendations only due to the network conditions, and over\_rec recommendations do not happen in this context. On the other hand, the network resource awareness of RAViR still provides a remarkable advantage. For example, RAViR offers an average QoE more than 100% higher than all JCRP variants in all evaluated scenarios and cache capacities.

Generally, the users of a Radio Access Network (RAN) experience the same *ICI* at a specific point in time, but each UE has its own wireless link, i.e., UEs may obtain different CQI values. Thus, CQI is relevant in the choice of the video representation and, as a consequence, in the QoE value. Fig. 9 shows the QoE values as a function of the CQI and video representation. If all representations of all videos were available in the cache system, given a certain CQI value for the UE wireless link, it is expected that a network-aware caching and recommendation method chooses the video representation that provides the highest QoE value (lighter colors in Fig. 9). On the other hand, caching and recommendation methods unaware of network resources may frequently choose video representations under or over the UE wireless link available bandwidth (darker colors in Fig. 9), which impacts negatively in the QoE. Since RAViR is aware of computing and network resources, it operates in the optimal range illustrated in Fig. 9.

## 6 CONCLUSIONS

In this paper, we focus on jointly optimizing cache and recommendations for improving network performance and user satisfaction. We present an optimization model that represents a recommendation system aware of the available resources (cache, resources of the users' devices, and network resources) and coupled with the cache storage policy. We observed noticeable improvements in user satisfaction by taking into consideration the device limitations, video representation, and network conditions in the joint decision on video recommendation and content caching. Additionally, we realized the inability of the commonly adopted cache hit ratio to properly capture important QoE aspects, which motivated the introduction of our QoE metric. Our proposal, named RAViR, outperforms a state-of-the-art (SOTA) method in all evaluated scenarios. RAViR increases user QoE by at least 68% and effective cache hit ratio by at least 14% in comparison with the JCRP variants, i.e., a SOTA method.

As future work, we intend to evaluate RAViR in a laboratory testbed, making recommendations and content caching online for synthetic users. In this type of environment, we can evaluate the effective cache hit ratio and also measure other QoE metrics, such as the ITU-T rec. P. 1203 [29]. A basic assumption for our model has been that the regular user keeps some time between the consumption of video content while she moves slowly inside the coverage of a base station. Thus, an extension would be to consider more dynamic mobility scenarios and users' content request patterns. In this respect, we are interested in investigating the coefficient of variation of CQI values observed in a sliding window (with a size lower than 1h) as decision-making for RAViR triggering. Finally, since 5G is reintroducing satellites as an effective part of the mobile networks infrastructure, this represents an opportunity for investigating the design of video content caching due to the several aspects involved. In the context of 5G non-terrestrial networks (NTNs), i.e., considering also satellites, caching may address bottleneck and also latency issues. Among the aspects that may impact the design of video content caching there are: the type of spaceborne device (e.g., GEO High Throughput Satellite or LEO satellite constellation), communication and computing capacities of the spaceborne devices, and type of satellite architecture (i.e., access or relay-like).

## ACKNOWLEDGMENTS

This work was partially funded by the Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG) in the call No. 03/2015 FAPEG/CNPq/PPP, process No. 201810267001734. This work was also supported by the Ministério da Ciência, Tecnologia e Inovação (MCTI) through the project "Brasil 6G", process No. 01245.010604/2020-14 and by project EM-BRACE Inria associated-team.

## REFERENCES

- [1] Cisco VNI, *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*, <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf>, 2020.
- [2] L. E. Chatzileftheriou *et al.*, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *IEEE Conference on Computer Comm. - INFOCOM*, 2017, pp. 1–9.
- [3] L. E. Chatzileftheriou *et al.*, "Jointly Optimizing Content Caching and Recommendations in Small Cell Networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 125–138, 2019.
- [4] Y. C. Hu *et al.*, *Mobile Edge Computing: A key technology towards 5G*, ETSI White Paper N. 11, 2015.
- [5] S. Kekki *et al.*, *MEC in 5G networks*, ETSI White Paper No. 28, 2018.
- [6] European Telecommunications Standards Institute (ETSI), *Multi-access Edge Computing (MEC); Radio Network Information API*, ETSI GS MEC 012 V2.1.1, 2019.
- [7] F. Malandrino *et al.*, *From Megabits to CPU Ticks: Enriching a Demand Trace in the Age of MEC*, 2020.
- [8] T. V. Doan *et al.*, "A longitudinal view of netflix: Content delivery over ipv6 and content cache deployments," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 1073–1082.
- [9] C. A. Gomez-Urbe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, 13:1–13:19, 2015.
- [10] P. Sermpetzis *et al.*, *Towards qos-aware recommendations*, 2020. arXiv: 1907.06392 [cs.MM].
- [11] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, Dec. 2015, ISSN: 2160-6455. DOI: 10.1145/2827872. [Online]. Available: <https://doi.org/10.1145/2827872>.

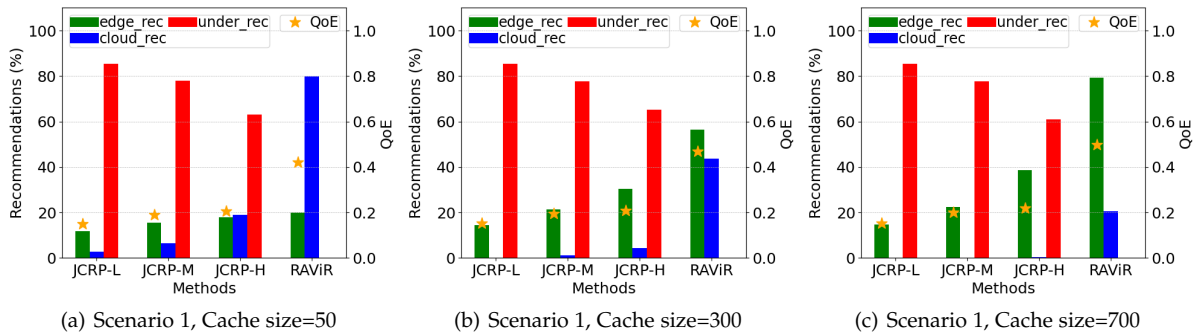


Fig. 8. Percentage of recommendations by type and achieved QoE, for each method, in Scenario 1 and total cache capacities of 50, 300 and 700.

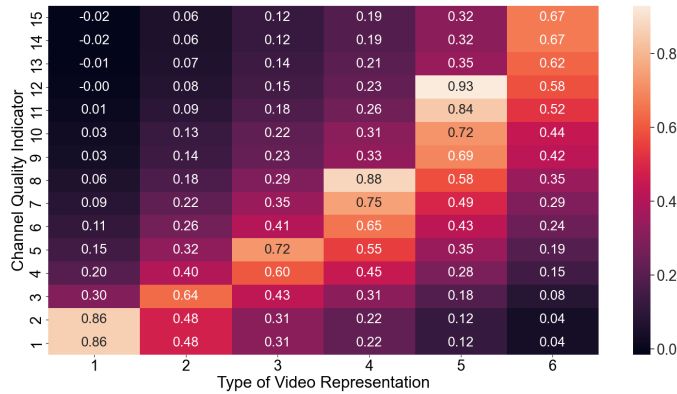


Fig. 9. QoE values as a function of CQI and video representation.

[12] W. Ali *et al.*, “A survey of web caching and prefetching,” *Int. J. Advance. Soft Comput. Appl.*, vol. 3, no. 1, pp. 18–44, 2011.

[13] I. U. Din *et al.*, “Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions,” *IEEE Communications Surveys & Tut.*, vol. 20, no. 2, pp. 1443–1474, 2018.

[14] D. K. Krishnappa *et al.*, “Cache-Centric Video Recommendation: An Approach to Improve the Efficiency of YouTube Caches,” *ACM Trans. Multimedia Comput. Comm. Appl.*, vol. 11, no. 4, 2015.

[15] S. Dernbach *et al.*, “Cache content-selection policies for streaming video services,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[16] P. Sermpezis *et al.*, “Femto-Caching with Soft Cache Hits: Improving Performance with Related Content Recommendation,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[17] J. Tadrous *et al.*, “Proactive Content Download and User Demand Shaping for Data Networks,” *IEEE/ACM Trans. on Networking*, vol. 23, no. 6, pp. 1917–1930, 2015.

[18] Z. Li *et al.*, “Toward A Practical Perceptual Video Quality Metric,” <https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652>, Accessed: March-10-2021, 2016.

[19] J. Nightingale *et al.*, “5g-qoe: Qoe modelling for ultra-hd video streaming in 5g networks,” *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 621–634, 2018.

[20] K. Saija *et al.*, “A Machine Learning Approach for SNR Prediction in 5G Systems,” in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2019, pp. 1–6.

[21] S. Yang *et al.*, “Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/qoe models,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1888–1902, 2019.

[22] D. De Vleeschauwer and K. Laevens, “Performance of caching algorithms for iptv on-demand services,” *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 491–501, 2009.

[23] S. Traverso *et al.*, “Temporal locality in today’s content caching: Why it matters and how to model it,” *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 5, pp. 5–12, 2013.

[24] M. H. Mohamed *et al.*, “Recommender systems challenges and solutions survey,” in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 2019, pp. 149–155. DOI: 10.1109/ITCE.2019.8646645.

[25] R. Zhou *et al.*, “The Impact of YouTube Recommendation System on Video Views,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010, pp. 404–410.

[26] V. Cacchiani *et al.*, “Knapsack problems — An overview of recent advances. Part I: Single knapsack problems,” *Computers & Operations Research*, vol. 143, pp. 1–14, 2022.

[27] P. Sermpezis *et al.*, “Soft cache hits: Improving performance through recommendation and delivery of related content,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1300–1313, Jun. 2018, ISSN: 1558-0008. DOI: 10.1109/JSAC.2018.2844983.

[28] M. Vora *et al.*, “Content Popularity for Open Connect,” Available at <https://netflixtechblog.com/content-popularity-for-open-connect-b86d56f613b>, Accessed: March-10-2021, 2017.

[29] W. Robitzta *et al.*, “HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software,” in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 466–471.

**Ana Claudia B. L. Monção** is an assistant professor at the Institute of Informatics - Universidade Federal de Goiás (UFG) since 2013. She holds a degree in Data Processing Technologist from Pontifical Catholic University of Rio de Janeiro (1992), has MSc in Computer Sciences from UFG (2013) and she is currently PhD candidate at UFG. Her main research interests include wireless networks, recommendation systems and data mining.

**Sand Luz Correa** is an associate professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where she has been a professor and researcher since 2010. She holds a degree in Computer Science from UFG (1994), has MSc (1997) in Computer Sciences from University of Campinas (UNICAMP) and PhD (2011) in Informatics from Pontifical Catholic University of Rio de Janeiro (PUC-Rio). In 2015, she spent her sabbatical at Virginia Tech (in the USA) and, in 2020, at Inria Saclay Research Centre (in France). Her research interests include cloud computing, SDN, data plane programmability, and smart city platforms.

**Aline Carneiro Viana** is a permanent Research Director (DR) at Inria, where she leads the team TRIBE team. After a 1-year sabbatical leave at the TKN Group of the TU-Berlin, Germany, she got her habilitation degree from UPMC - Sorbonne Universités, France in 2011. Dr. Viana got her PhD in Computer Science from the UPMC - Sorbonne Universités in 2005. Her research addresses the design of solutions for tactical networking, smart cities, mobile and self-organizing networks with the focus on human behavior analysis. She is a recipient of the French Scientific Excellence award since 2015 and for 6 years now and was nominated in 2016 as one of the “10 women in networking/communications that you should Watch”.

**Kleber Vieira Cardoso** is an associate professor at the Institute of Informatics - Universidade Federal de Goiás (UFG), where he has been a professor and researcher since 2009. He holds a degree in Computer Science from Universidade Federal de Goiás (1997), has MSc (2002) and PhD (2009) in Electrical Engineering from COPPE - Universidade Federal do Rio de Janeiro. In 2015, he spent his sabbatical at Virginia Tech (in the USA) and, in 2020, at Inria Saclay Research Centre (in France). His research is focused on the following topics: wireless networks, SDN, virtualization, resource allocation, and performance evaluation.