Ad Hoc Networks 89 (2019) 170–176

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

QoE-driven resource allocation for massive video distribution☆

Luca De Cicco*, Saverio Mascolo, Vittorio Palmisano

Dipartimento di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari, Via Orabona 4, Bari 70125, Italy

ARTICLE INFO

Article history:
Available online 7 March 2019

Keywords:
Adaptive video streaming
Video Control Plane
Quality of Experience

ABSTRACT

Massive video delivery systems employ the HTTP protocol and multiple Content Delivery Networks (CDNs), which serve the content to the end-users on behalf of the video providers and guarantee scalability and Quality of Experience (QoE). In this paper, a Video Control Plane (VCP) is presented which monitors the QoE delivered by any of the CDN belonging to its pool and selects the most performing one when a new video request is received. The VCP employs a continuously updated prediction of the CDNs performances based on the feedback sent by the video clients and computed through a k-NN regression algorithm. The proposed VCP has been evaluated through simulations and shows significant performance improvement in terms of QoE delivered to the user.

1. Introduction

The amount of video content that is being distributed over the Internet is increasing due to the wide diffusion of Smart TVs, tablets, and smartphones [2]. The growth of video traffic poses new challenges to video providers. On the one side, they strive to increase the Quality of Experience (QoE) perceived by the end users [3–5], which is a crucial factor influencing user engagement and thus revenues [4]. On the other side, they are interested in minimizing the costs of the delivery infrastructure that is responsible for serving millions of concurrent viewers. Scalability of video delivery systems is guaranteed by Content Delivery Networks (CDNs), which are globally distributed networks of HTTP servers deployed in multiple data centers. CDNs serve the content to the end-users on behalf of the video providers, which are billed with "pay-as-you-go" pricing plans based on the streamed egress traffic. After leaving the CDN, the video content is delivered to the end-user through his access Internet Service Provider (ISP) network, which in some cases might be interconnected with peered ISP providers. The end-to-end delivery path may get congested and the decreased available bandwidth may severely affect the QoE. In order to address such issues, the mainstream approach is that of the HTTP Adaptive Streaming (HAS) which requires the video players, running at the client, to implement adaptive streaming control algorithms. Such algorithms can dynamically adapt the video bitrate based on the estimated network conditions to maximize the QoE, which is significantly impacted by playback interruptions (rebuffering events) and low average video quality [5]. As such, the goal of adaptive streaming algorithms is to avoid rebuffering events and possibly matching the available bandwidth, which is limited by the CDN egress bandwidth and the ISP network bandwidth.

It has been shown that CDNs performances are subjected to significant variability, both temporally and spatially [6,7]. The temporal variability of the CDNs performance is due to the time-varying CDN load, which is unknown to video providers. In extreme cases, the so-called*flash-crowd events*, the CDN may suffer from a severe performance degradation. The spatial variability is due to factors such as the geographical regions and ISP to which the user is connected. Users downloading a video from the same CDN but through different ISPs can experience remarkably different performances. Performance variability due to client devices and players can also be included in this category. Research has mainly focused on addressing such issues by designing control algorithms to manage internal CDN resources, such as [8–10].

In this work, we propose a control algorithm to dynamically assign the optimal CDN based on a QoE prediction algorithm. We assume that a video provider has a pool of candidate CDNs available for serving the same video content. We have designed a Video Control Plane (VCP) allowing the video provider to monitor the QoE delivered by each CDN and selecting the most performing one for each new video request. In particular, the second task is performed by a resource allocation (RA) module. This module employs a continuously updated prediction of the CDNs performances based on the feedbacks sent by the video clients. This is required by the fact that video providers are not able to either directly monitor the

☆ This is an extended version of the paper [1].

* Corresponding author.
E-mail addresses: luca.decicco@poliba.it (L.D. Cicco), mascolo@poliba.it (S. Mascolo), vpalmisano@gmail.com (V. Palmisano).

https://doi.org/10.1016/j.adhoc.2019.02.008
1570-8705/

resource states or control resource allocation inside the CDN. The proposed VCP employs a modified version of the *k*-Nearest Neighbors (*k*-NN), a well-known algorithm employed in machine learning and regression domains. Differently from other works proposed in the literature, such as [6], we have taken a decoupled approach where the video bitrate adaptation is performed independently at the client without assistance from the VCP. This approach has several important advantages, such as the decrease of communication exchange between VCP and clients, design and implementation scalability, compared to the one where the VCP also selects the video bitrate for each client. We have evaluated the performance of the proposed VCP by means of numerical simulations, focusing on the QoE prediction algorithm.

The rest of the paper is organized as follows: Section 2 summarizes the related work; in Section 3 we describe the proposed VCP architecture; Section 4 presents the QoE prediction algorithm employed by the VCP; Section 5 discusses the results and Section 6 concludes the paper.

2. Related work

This section provides a review of the literature focusing on resource allocation strategies for video streaming services. In particular we cluster the related work into two categories: (1) studies focusing on the optimal selection of CDNs; (2) papers addressing the resource allocation problem in the Cloud.

2.1. Optimal selection of CDNs

In [11] the video delivery system employed by Hulu and Netflix, two popular online video services, has been studied through active measurements. The work focuses on how Hulu and Netflix select CDNs and how each CDN allocate resources (servers) to serve user requests. Extensive data collection, analysis, inference and systematic experimentation have been conducted to the purpose. The main finding is that Hulu distributes user requests among the CDNs according to a certain predetermined ratio. The selection of the *preferred* CDN for a given user does not seem to take into account the current network performance between the user and the selected CDN. Moreover, Hulu often varies preferred CDNs for each user. Once a CDN is selected, Hulu clients are typically assigned with the same CDN during the entire length of the session even when performance of the CDN degrades. The study reveals that the CDN is only changed when it is not able to provide the lowest quality level of the user. Regarding CDNs, authors find that CDNs employ varying number of servers at different locations to provide Hulu content. The same study describes the basic architecture of the Netflix video streaming platform. This is done by monitoring the communication between the Netflix player and several components of the Netflix platform. A large number of Netflix video streaming manifest files is collected to analyze how geographic locations, client capabilities, and content type impact on the streaming parameters used by Netflix, such as content formats, video quality levels, CDN ranking. The main finding is that Netflix players remain attached to a fixed CDN even when the other CDNs can provide better video quality. An extensive bandwidth measurement study of the three CDNs used by Netflix is performed. The paper shows that CDN performance changes significantly across time and location. In addition some alternative mechanisms to improve video delivery performance by using multiple CDNs are proposed. In particular, it is shown that selecting the best serving CDN based on a small amount of measurements at the beginning of each video session can provide more than 12% bandwidth improvement over the static CDN selection strategy employed by Netflix. Using multiple CDNs simultaneously, an improvement of more than 50% can be achieved.

Compared to this work, this paper proposes a QoE-aware resource allocation that takes into account the ISP the user is connected to improve QoE prediction.

A comprehensive analysis of such an issue is performed in [6]. The study starts from the observation that there is a mismatch between the requirements of video streaming and the architecture of today' s HTTP-based video delivery infrastructures, both at the ISP and CDN level. Collecting fine-grained client-side measurements from more than 200 million client viewing sessions, authors find that 20% of the sessions are affected by a rebuffering ratio of more than 10%, 14% of users wait more than 10 seconds for video to start up, more than 28% of sessions have an average bitrate less than 500 kbps, and 10% of users fail to see any video at all. Authors find that the causes of these performance problems are: (1) the significant spatial diversity in CDN performance and availability across different geographical regions and ISPs; (2) the substantial temporal variability in the CDN performance and client-side network performance; (3) poor system response to overload scenarios when flash crowd events occur in particular regions or ISPs.

In [12] the issue of the optimal CDN assignment is tackled by using two algorithms: (1) an optimization algorithm executed at the content publisher to guide content distribution; (2) a simple algorithm executed at content viewers to execute local adaptation. Results show that the proposed algorithms reduce publishing cost by up to 40%. Moreover, according to the experimental results client algorithm is able to reduce QoE degradation by 51%.

Compared to this study, our paper proposes a control plane decoupling the problem into two subproblems: the first one, solved at the clients, strive to adapt to changing network bottleneck available bandwidth by means of the adaptive video streaming algorithm; the second one, executed at the Video Control Plane, allocates the user to the CDN that is expected to provide the best QoE based on a prediction algorithm. Our proposed approach has the advantage of being orthogonal to the implemented adaptive video streaming strategy. The only additional requirement is that clients report some QoE-related feedback to the control plane.

2.2. Optimal resource allocation in the cloud

The design of an effective control plane for the automatic resource allocation of a cloud is an important open research topic. In [13], by employing time series forecasting methods to predict the user demand, a bandwidth resource reservation algorithm, along with a load balancer, is designed. In particular, a bandwidth auto-scaling facility that dynamically reserves resources from multiple data centers for VoD providers, with several distinct features, is proposed. The facility tracks the history of bandwidth demand in each video channel using cloud monitoring services, and periodically estimates the expectation, volatility and correlations of demands in all video channels for the near future using time-series techniques. Moreover, quality assurance is provided by judiciously deciding the minimum bandwidth reservation needed to satisfy the demand with high probability. The bandwidth minimization problem given the predicted demand statistics as input is formulated, the theoretically optimal load direction across data centers is derived, and heuristic solutions that balance bandwidth and storage costs are proposed. Extensive trace-driven simulations based on a large data set of 1693 video channels collected from UUSee, a production VoD system, over a 21-day period are used to evaluate the proposed framework. In [14] a model for elastic media streaming service over a public Cloud is presented: authors propose the Virtual Content Service Provider (VCSP) which has the role of dynamically renting cloud resources by cloud service providers to adjust the bandwidth resources to the user demand. The algorithm takes explicitly into account the minimization of distribution costs by deciding which type of instance to rent, i.e. small, large,

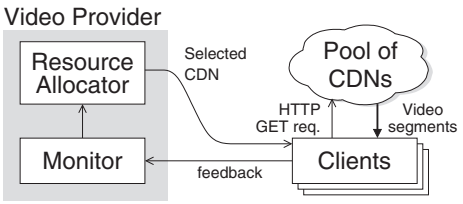

Fig. 1. Architecture of the proposed Video Control Plane.

spot. A theoretical model to explore the trade-off between the procurement cost and the achieved user QoE for cloud-based video streaming is developed. The problem is formulated as a joint optimization problem of resource provisioning and procurement under price variety and demand dynamics. An approximate online algorithm, called OPT-ORS, is proposed with an explicitly provable performance upper bound. Low complexity is ensured by exploiting the structural properties of the optimal solution. Extensive trace-driven simulations have been conducted to verify its effectiveness. In [10] proposed a central controller for resource allocation (RAC) which automatically throttles the number of active machines in a Cloud-based adaptive streaming delivery system to adapt to the user workload with the goal of obtaining a high video quality while minimizing delivery costs. The proposed controller has been implemented in a discrete event simulator and, using a realistic workload, a comparison with two other controllers has been carried out. The results have shown that the proposed controller is able to deliver a high video quality to the users while containing the delivery costs. In particular, it has been shown that the proposed controller is able to save 33%, 28%, and 10% of the distribution costs compared to the static controller, the feed forward controller, and a version of the RAC without predictor respectively.

Compared to the described works, the present paper proposes an algorithm to optimally assign users to a CDN among the pool of available CDNs. The proposed system does not need to manage internal CDN resources, but instead tries to estimate the best performing CDN in terms of expected QoE for each user.

3. The control architecture

3.1. The architecture of the Video Control Plane

The proposed Video Control Plane (VCP) takes a decoupled design approach which separates the tasks of CDN assignment and bitrate adaptation. The VCP is responsible for selecting the CDN to each new user video request, whereas each client is responsible for selecting independently the video bitrate based on its current state. In particular, the video provider employs the Dynamic Adaptive Streaming over HTTP (DASH) standard to deliver videos to the users through standard HTTP servers deployed in the CDNs [15]. According to the DASH standard, videos are encoded into a number of *representations* which are characterized by different bitrates and video resolutions. Each video representation is then divided temporally into *segments* (or *chunks)* of fixed duration. Typical representation bitrates vary from few kbps (for low resolutions) up to several Mbps (f.i, in the case of 4K resolution). In DASH compliant video streaming systems, the client implements in the video player an adaptive streaming control algorithm to decide for each segment which representation to be downloaded in order to adapt to the time-varying end-to-end Internet bandwidth and avoid playback interruptions due to buffer depletion.

The overall architecture of the proposed VCP is shown in Fig. 1. The VCP is made of (1) a *Monitor*; (2) a *Resource Allocator* (RA); (3) a pool of *Resources* that the streaming platform employs to stream the videos to the users; (4) *Measurement* modules placed

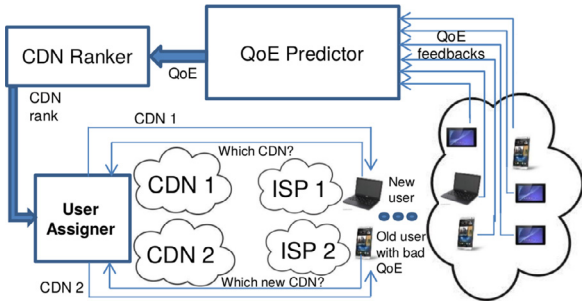

Fig. 2. The architecture of the resource allocation module.

at the clients in the video players that send feedback signals. The goal of the control plane is to provide a system that automatically selects a CDN for each video request issued by the user with the aim of providing the best Quality of Experience (QoE). Notice that, as shown in Fig. 1, once the Resource Allocator selects the best performing CDN in the CDN pool, the client will directly issue HTTP GET requests to fetch video segments from the CDN suggested by the RA.

In the following we present the essential features of each component:

- the *Measurement module* runs on the video players at the clients and periodically sends measurements through feedback messages to the *Monitor* module;
- the *Monitor module* processes and stores the feedback messages received by the video players*Measurement* modules; the monitor responds to *Resource Allocator* queries to provide either raw or aggregated measurements;
- the *Resource Allocator* is the module where decisions are made regarding the allocation of the resources with the aim of maximizing the QoE;
- the *Resources* are not part of the streaming platform and are mainly composed of (1) storage resources where available videos are placed and (2) network resources that are used to deliver the video to the end-users.

3.2. The resource allocation module

The goal of the resource allocation module is to assign users requests to CDNs based on QoE predictions. Fig. 2 shows the overall control architecture of the resource allocation module consisting of three blocks: the QoE Predictor, the CDN Ranker, and the CDN Assigner. We recall that this module is placed at the video provider as shown in Fig. 1.

The resource allocation module executes two tasks: *CDN ranking* and *CDN assignment*. The CDN ranking is periodically executed every T_s and is responsible for: (1) computing the updated prediction of the QoE for each CDN; (2) ranking the arrays of CDNs for each class of users based on the predicted QoE. The CDN assignment is executed upon the arrival of a new user request and is responsible for selecting the first CDN of the ranked array corresponding to the class of the user. The first task is executed by the QoE Predictor and CDN Ranker. The second task is executed by the User Assigner.

QoE Predictor. This module predicts, for a given future temporal horizon, the QoE that will be delivered to new users based on past and current QoE measurements. The QoE Predictor computes the expected QoE a given user would obtain with each CDN candidate. To the purpose, users are grouped by classes, as detailed in Section 3. The module takes as input the feedbacks which are periodically sent by the clients and gives as output the predicted QoE in the next temporal horizon. User' s feedback includes the player

status (play or pause), the playout buffer length measured in seconds, the bandwidth estimate measured in kb/s, and the selected video level bitrate measured in kb/s. These variables are then employed to compute the metrics allowing to measure the QoE perceived by users. Among the metrics that are commonly considered in the literature [5,7,16–18], we consider the rebuffering ratio, the average bitrate, the start-up delay, the amplitude and frequency of video level switches. Notice that today it is commonplace for video clients to periodically send feedback information via HTTP GET requests to the video provider (e.g., YouTube, Netflix). The added overhead to handle feedback from clients is low considering that: (1) video servers already handle HTTP GET requests issued from the clients to fetch video segments; (2) the frequency of feedback is typically lower (in the order of minutes) compared with the frequency of HTTP GETs issued to fetch video segments (in the order of 2–10 s). Thus, the use of such feedback information does not pose significant implementation or scalability issues with respect to simpler algorithms which are not QoE-aware but allows to take more informed decision on resource allocation.

CDN Ranker. This module ranks the available CDNs for each users class. The ranking is based on the predicted QoE, provided by the QoE predictor module.

User Assigner. This module assigns a CDN to a given user when a video session is started. The User Assigner serves as the user front-end. It takes as input the CDN rank from the CDN Ranker. Its output is the CDN assigned to the connection request. When the connection request is received, this module extracts from the CDN rank the best CDN for the given user and performs the assignment.

4. The QoE prediction algorithm

This section describes the proposed algorithm to predict the QoE delivered by each CDN candidate to a newly arrived user, which can be cast to a classic regression problem. The key idea that we leverage is that the prediction of the value (i.e., the QoE experienced by the user) assumed by an object (i.e., the new video session) is made based on the values assumed by similar objects (past and current video sessions) in a close temporal horizon. The accuracy of such a prediction ultimately depends on the estimated similarity between the newly arrived video session and the running video sessions that are employed for the prediction. Similarity between video sessions can be estimated based on factors impacting the QoE, which have been identified in [6]: the CDN streaming the video, the ISP to which the client is connected, the peering between the ISP and the CDN, the geographical region where the client is located, the kind of connection (WiFi, 3G, 4G, ADSL, fiber), the client device and the client player. The higher the similarity of these factors between two video sessions, the higher the prediction accuracy.

The proposed QoE Prediction module employs a modified version of the well-known *k-Nearest Neighbors* (k-NN) [19,20], a simple yet powerful algorithm that is used in several applications in the context of machine learning, classification, and regression. In machine learning contexts the aim is to find the class to which an object belongs based on its characteristics. The key idea behind k-NN is to assign the unknown object to the class to which the majority of its k most similar neighbors belong. Similarity between objects is measured through a distance function, such as the Euclidean, the Minkowski, or the Hamming distance, which takes as input the characteristics (or attributes) of the object. In regression contexts, where the aim is to predict the expected value an object will assume, the value of the unknown object is expected to be equal to the weighted average of the k most similar objects. Weights are usually set as the inverse of the distance. The neighbors are taken from a set of objects for which the value is already known.

Table 1
Matrix of the predicted QoE Q.

	CDN 1	CDN 2	CDN 3
ISP 1	$q_{11} = 0.9$	$q_{12} = 0.5$	$q_{13} = 0.95$
ISP 2	$q_{21} = 0.5$	$q_{22} = 0.7$	$q_{23} = 0.8$
ISP 3	$q_{31} = 0.8$	$q_{32} = 0.6$	$q_{33} = 0.5$

In our implementation we consider two attributes, the CDN and the ISP. We did not consider secondary attributes, such as the type of Internet connection, the client device features, since with our decoupled approach a fine-grained control loop executing bitrate adaptation runs at the clients. We have employed the Hamming distance, i.e. the number of attributes at which the corresponding values are different, to measure the similarity between different video sessions. With two attributes the Hamming distance between two different video sessions can assume only three possible values: 0 (same ISP and same CDN), 1 (either CDN or ISP is different), and 2 (both ISP and CDN are different). For this reason, we can cast our regression problem to the following one.

We define a *user class* as the set comprising users having a distance equal to 0 from each other, i.e. belonging to the same pair (ISP, CDN). In particular, if we denote with M the number of ISPs and by N the number of available CDNs in the pool, we can define $M \times N$ user classes. We denote the user class $\mathcal{C}_{ij}$, $i = 1, \ldots, M$, $j = 1, \ldots, N$, as the set of users belonging to the ith ISP and assigned to the jth CDN. Next, for each user class $\mathcal{C}_{ij}$, we can compute the average QoE q_{ij} by considering the QoEs obtained by all users belonging to such class. Thus, at each execution step of the prediction algorithm, the $M \times N$ matrix $Q = [q_{ij}] \in \mathbb{R}^{M \times N}$ can be computed. Finally, we predict the QoE a new video request coming from ISP provider i would perceive if the candidate CDN j were selected by simply extracting the element q_{ij} from the matrix Q. Finally the user is assigned to the CDN whose predicted QoE is the best possible. Notice that the way the QoE is measured depends on the explicit feedback provided by the user to the Monitor module (see Fig. 1). The actual functional employed to measure the QoE of a video session is implementation-dependent.

Table 1 shows an example of a possible situation where three CDN candidates are available and users are connected through three different ISPs. If, f.i., a new user connected through ISP 2 starts a video session, the proposed algorithm would assign it to the best performing CDN, that in this example is the CDN 3 having the best QoE for that particular ISP. Thus, in general, a user connected through the ISP I is assigned to the Jth CDN, with J computed as follows:

$$J = \underset{j \in \{1,\ldots,N\}}{\operatorname{argmax}} \; q_{Ij}.$$

Some further issues have to be taken into account. The QoE prediction of a given class $\mathcal{C}_{ij}$ is statistically significant if a sufficient number of video sessions belongs to that class, i.e. the number of users in $\mathcal{C}_{ij}$ gets above a significance threshold. To the purpose, a round robin algorithm is employed to let each candidate CDN reach the target threshold before using the CDN Assigner driven by the QoE Prediction.

5. Results

In this section we provide a performance evaluation of the proposed resource allocation strategy by employing a simulator that we have developed in Matlab. In particular, Section 5.1 describes the simulation setup, whereas Section 5.2 provides an analysis of the obtained results.

5.1. Simulation setup

The proposed Video Control Plane has been implemented in a simulator written in Matlab. The proposed dynamic CDN assignment strategy based on the *k-NN* prediction technique has been compared to other algorithms (see Section 5.1.2).

5.1.1. The scenario

The scenario we consider is composed of one video provider which streams a live video event by using a pool of CDNs to a large number of users watching the event. In this paper, the video generated by the video producer is encoded into five representations at bitrates $r_i \in \mathscr{R} = \{300, 900, 1500, 2500, 3500\}$ kbps. These are commonplace bitrate values employed by video streaming platforms supporting resolution ranging from 320×240 up to 1920×1080. Moreover, in this paper we consider each representation divided into segments of duration equal to 5 s, which is a duration typically used by video streaming platforms such as Netflix, YouTube, and Akamai [21]. In this paper, the algorithm *ELASTIC* [22,23] is used at the client to control the playout buffer and dynamically select the video bitrate. The algorithm is designed using a feedback control technique known as *feedback linearization* and is fully compliant to the DASH standard [22]. In particular, the algorithm decides the video representation to be downloaded according to the following control law:

$$r_k = Q\left(\frac{\hat{b}_k}{1 - K_1 q_k - K_2 q_k^I}\right)$$

where (1) $r_k \in \mathscr{R}$ is the video representation chosen by the controller for the download of the kth segment measured in kbps; (2) q_k is the playout buffer level measured in seconds; (3) q_k^I is the integral of the error $q_T - q_k$; (4) q_T is the playout buffer target measured in seconds; (5) $\hat{b}_k$ is the bandwidth estimate measured in kbps; (6) $Q : \mathbb{R} \to \mathscr{L}$ is the quantization function returning the maximum video representation bitrate $r \in \mathscr{R}$ lower than its input value; (7) K_1 and K_2 are respectively the proportional and integral gain.

In the simulations we consider that the video producer employs a pool of $N = 3$ CDNs streaming a live event lasting three hours to a number of 10,000 users. The workload is generated as follows. Each user starts a streaming session at a random starting time during the event and remains connected until the end of the event. The starting time of the video sessions is uniformly distributed over the first 148 min of the whole video event duration. We assume that users are connected through three ISPs, i.e., $M = 3$ ISPs. Each user is randomly assigned to one of the ISPs according to a uniform distribution at the beginning of the simulation. The QoE prediction is executed at each discrete time instant t with a sampling interval T_s. The sampling time T_s employed in the simulations has been set equal to 5 min.

Let us now describe how the available bandwidth for each video session s is assigned during the simulation. At the tth simulation step, the available bandwidth $b_s(t)$ each video session s receives is computed as the sum of two terms:

$$b_s(t) = b_{ij}(t) + n_s(t).$$

$b_{ij}(t)$ is the bandwidth component due to the peering between ISP i and CDN j assigned to the video session, whereas $n_s(t)$ is the user-specific component modeling the impact on the video session of factors such as type of the client connection and client location. The component $b_{ij}(t)$ is extracted from the $B(t)$ matrix, which models the bandwidth patterns of all the possible pairs (ISP, CDN) at the tth time instant. Notice that the Video Control Plane does not have any knowledge on the entries of $B(t)$ since in this work we make the realistic assumption that there is no cooperation between CDNs and the Video Control Plane. The user-specific component $n_s(t)$ is extracted from a normal distribution with zero mean and standard deviation equal to 150 kbps.

5.1.2. The resource assignment strategies

In order to assess the performance of the resource assignment of the proposed strategy, we have compared the following four algorithms.

Oracle *(benchmark).* The CDN assignment is based on the exact a-priori knowledge of the average bandwidth value for each pair (ISP, CDN), i.e. on the knowledge of the series of matrices $B(t)$. This algorithm is employed as a benchmark to compare the different CDN assignment strategies. Obviously, this algorithm is not implementable since in practice it is not possible to have an exact knowledge of $B(t)$ without having a cooperation between CDNs and ISPs. As such, no other algorithm can perform better than the *Oracle*.

k-NN. This is the proposed QoE-aware strategy that has been described in Section 3.

Global. This assignment strategy is based on the QoE prediction obtained by computing the average QoE for each CDN candidate, without taking into account the ISP. Such an approach, compared to k-NN, is simpler to be implemented. In particular, this strategy only requires to group users' feedbacks based on the CDN assigned to them. Then, the average QoE is computed for each CDN based on such feedbacks. Finally, CDNs are ranked bsed on the average QoE and users are directed to the best performing CDN.

Round-robin. This CDN assignment is not based on QoE predictions. A simple round-robin algorithm uniformly balances the user requests among the N CDN candidates. Due to its simplicity, this approach is widely used in the industry to load-balance among CDN candidates. The clear advantage of round-robin is its implementation simplicity since it does not need any feedback from users and no control plane is needed. We have considered this strategy as the baseline for QoE-aware strategies.

5.1.3. Metrics

In this work, we measure the QoE as proposed in [7], i.e.:

$$QoE(Q, R) = Q - R \tag{1}$$

Eq. (1) gives a QoE score which depends on two terms:

1. The rebuffering penalty $R = \alpha \cdot RR$, where RR is the *Rebuffering Ratio*, which is the most important factor impairing the perceived QoE [5,16]. The rebuffering ratio is computed as the ratio between the total time the player has been paused due to buffer depletion and the total duration of the video streaming session.
2. The average video quality $Q = \beta \cdot AB$, where AB is the *average bitrate*, which is the second most important parameter impacting the perceived QoE [5,16]. The higher the average bitrate, the higher the visual quality experienced.

This functional has been obtained with a regression technique from a very large dataset containing performance measurements of millions of video sessions in the Internet. In particular, the weight α for the rebuffering penalty has been found to be equal to 3.7, whereas the constant β which weights the average bitrate is equal to 1/20. The QoE functional (1) expresses that the lower the rebuffering ratio and the higher the average bitrate, the better the user's perceived quality. Notice that, unfortunately, those two terms are conflicting. In fact, if the adaptive video streaming controllers selects a too high video bitrate it is very likely that the

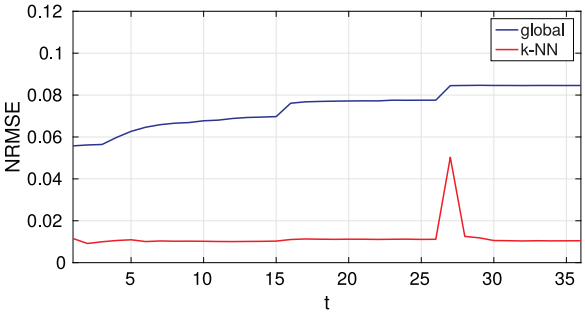

Fig. 3. NRMSE for the assignment algorithms based on QoE prediction, *k-NN* and *global*.

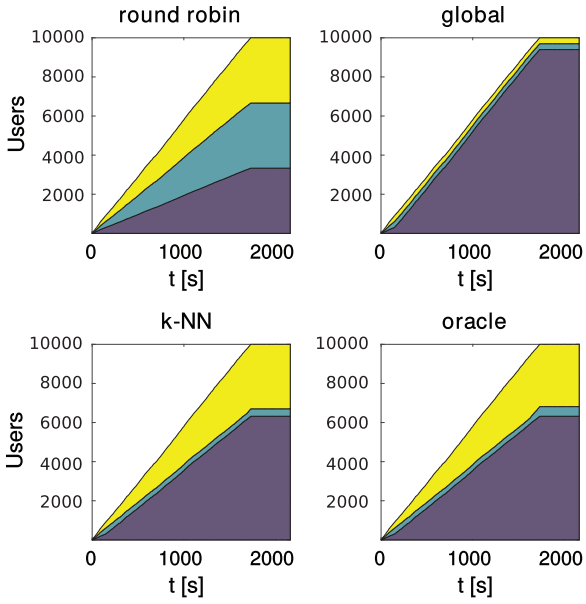

Fig. 4. Evolution over time of the CDN assignment.

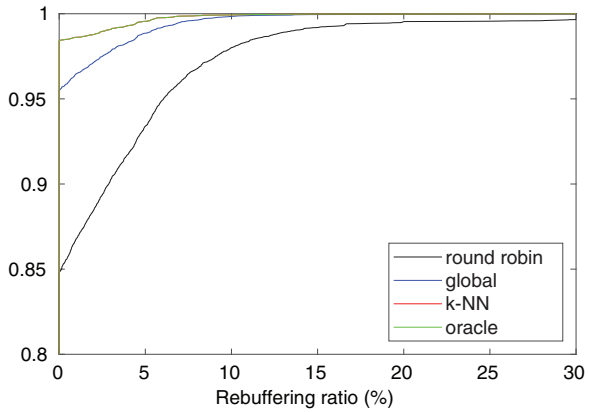

Fig. 5. CDFs of the rebuffering ratio *RR*.

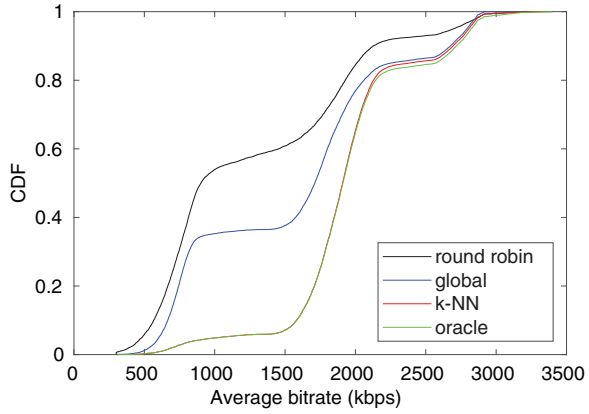

Fig. 6. CDF of the average bitrate *AB*.

rebuffering episodes would occur, ultimately affecting the overall QoE.

Finally, in order to measure the QoE prediction accuracy, we have employed the *Normalized Root Mean Square Error* (NRMSE), given by:

$$NRMSE = \frac{\sqrt{\frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N}(q_{ij}-\hat{q}_{ij})^2}}{q_{\max}},$$

where q_{ij}, $\hat{q}_{ij}$, $q_{\max}$ are respectively the predicted, the measured, and the maximum possible QoE. Notice, that a zero *NRMSE* indicates a perfect QoE prediction.

5.2. Results

We start by investigating the accuracy of the proposed QoE prediction strategy by measuring the NRMSE, i.e. the distance between the QoE predicted by the algorithm and the measured QoE. Fig. 3 compares the time evolutions of the NRMSE in the case of the *global* prediction strategy and the proposed *k-NN*. Notice that the round robin strategy is not included in this comparison since no QoE prediction is made when using this strategy. The figure clearly shows that the *k-NN* strategy outperforms the *global* predictor since it provides a smaller NRMSE.

Let us now compare the CDN assignment obtained when using the considered resource allocation algorithms. Fig. 4 shows the dynamics of the fractions of per-CDN assigned users of the two algorithms. Recall that, since *Oracle* has perfect knowledge of the matrix $B(t)$ at any given time, it provides the optimal assignment. Notice that, given the matrix $B(t)$ employed in this scenario, *Oracle* assigns a small fraction of users to the CDN 2. *k-NN*, the proposed strategy, exhibit a very similar behavior, confirming the accuracy of the *k-NN* predictor. *Global* assigns almost all the users to the single best performing CDN (CDN 1), without taking into account possible poor performance of the selected CDN with some ISPs. *Round-robin* performs a balanced assignment. *Oracle*, *k-NN*, and *global* keep a minimum fraction of the users assigned to each CDN candidate to enable performance monitoring at each time instant.

We next investigate how the considered CDN assignment strategies perform in terms of users' perceived QoE. The Rebuffering Ratio *RR* is shown in Fig. 5. *Oracle*, *k-NN* and *global* algorithms are able to keep *RR* below 1% for more than 95% of the users, clearly outperforming the *round-robin* assignment algorithm. This result shows that using a real-time estimate of the QoE can be beneficial and improve the overall QoE. Observe that *k-NN* obtains the roughly the same QoE of *Oracle* and provides better performance compared to *round-robin* and *global*. Interestingly, *global* provides better results with respect to *round-robin*. This confirms the need for an accurate prediction algorithm to be guaranteed that the performance of the QoE-driven assignment is higher than the one obtained with simple QoE-agnostic assignment schemes such as *round-robin*.

We now focus on the measured average bitrate *AB* which is a key parameter that relates to the perceived visual quality. Fig. 6 compares the measured *AB* obtained when using the considered strategies. *Round-robin* provides an *AB* higher than 2000 kbps to only 20% of the users. The *global* strategy delivers an average bi-

trate higher than 2000 kbps to 40%, whereas the proposed strategy *k-NN* further improves performances and is able to provide a bitrate higher than 2000 kbps to more than 60% of the users. This result clearly shows that the assignment algorithm as a high impact on the obtained average bitrate.

To conclude, we have found that the *k-NN* algorithm is remarkably more accurate than the *global* algorithm in predicting the QoE and obtains similar performance as the *oracle* benchmark. As a consequence, the proposed strategy is able to improve the overall QoE, in particular by increasing the obtained average bitrate and by decreasing the rebuffering ratio of video sessions.

6. Conclusions and future work

In this work, we have proposed a Video Control Plane to monitor the QoE delivered to video users by several streaming CDNs and allow the video provider to dynamically assign CDNs to the users based on QoE prediction. The CDN assignment is based on the k-NN regression algorithm, which improves the QoE prediction accuracy by identifying confounding factors such as poor performance of the ISP. The proposed assignment algorithm has been evaluated through numerical simulations. Results have shown that the NRMSE, assessing the accuracy of the employed regression algorithm, has been significantly improved with respect to the other considered algorithms. Moreover, both the rebuffering ratio and the average bitrate, key QoE factors, have been significantly improved. Future work will analyze the performance of the proposed control plane when also CDN costs are taken into account.

References

[1] G. Cofano, L. De Cicco, S. Mascolo, V. Palmisano, Qoe-driven resource allocation for massive video distribution, in: Proceedings of the First International Balkan Conference on Communications and Networking (BalkanCom'17), 2017.

[2] Cisco, Cisco Visual Networking Index:Forecast and Methodology 2013–2018 (2013).

[3] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, H. Zhang, Developing a predictive model of quality of experience for internet video, in: Proceedings of the ACM SIGCOMM, 2013, pp. 43,339–350.

[4] S.S. Krishnan, R.K. Sitaraman, Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs, IEEE/ACM Trans. Netw. 21 (6) (2013) 2001–2014.

[5] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, P. Tran-Gia, A survey on quality of experience of HTTP adaptive streaming, IEEE Commun. Surv. Tutor. 17 (1) (2015) 469–492.

[6] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, H. Zhang, A case for a coordinated internet video control plane, in: Proceedings of the ACM SIGCOMM, 2012, pp. 359–370.

[7] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, H. Zhang, Understanding the impact of video quality on user engagement, in: Proceedings of the ACM SIGCOMM, 2011, pp. 362–373.

[8] G. Mencagli, M. Vanneschi, E. Vespa, Control-theoretic adaptation strategies for autonomic reconfigurable parallel applications on cloud environments., in: Proceedings of the IEEE HPCS'13, 2013, pp. 11–18.

[9] D. Niu, C. Feng, B. Li, Pricing cloud bandwidth reservations under demand uncertainty, in: Proceedings of the ACM SIGMETRICS, 2012, pp. 151–162, doi:10.1145/2254756.2254776.

[10] L. De Cicco, S. Mascolo, D. Calamita, A resource allocation controller for cloud-based adaptive video streaming, in: Proceedings of the IEEE ICC, 2013, pp. 723–727, doi:10.1109/ICCW.2013.6649328.

[11] V.K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z.-L. Zhang, M. Varvello, M. Steiner, Measurement study of Netflix, Hulu, and a Tale of Three CDNs, IEEE/ACM Trans. Netw. 23 (6) (2015) 1984–1997, doi:10.1109/TNET.2014.2354262.

[12] H.H. Liu, Y. Wang, Y.R. Yang, H. Wang, C. Tian, Optimizing cost and performance for content multihoming, in: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM, 2012, pp. 371–382.

[13] D. Niu, H. Xu, B. Li, S. Zhao, Quality-assured cloud bandwidth auto-scaling for video-on-demand applications, in: Proceedings of the IEEE INFOCOM, IEEE, 2012, pp. 460–468.

[14] J. He, Y. Wen, J. Huang, D. Wu, On the cost–QoE tradeoff for cloud-based video streaming under amazon EC2's pricing models, IEEE Trans. Circuits Syst. Video Technol. 24 (4) (2014) 669–680.

[15] I. Sodagar, The MPEG-DASH standard for multimedia streaming over the Internet, IEEE MultiMed. 18 (4) (2011) 62–67.

[16] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, H. Zhang, Developing a predictive model of quality of experience for internet video, in: Proceedings of the ACM SIGCOMM, 2013, pp. 339–350, doi:10.1145/2486001.2486025.

[17] T. Hossfeld, M. Seufert, C. Sieber, T. Zinner, Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming, in: Proceedings of the QoMEX, 2014, pp. 111–116.

[18] T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, Quantification of YouTube QoE via crowdsourcing, in: Proceedings of the IEEE MQoE, 2011.

[19] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Stat. 46 (3) (1992) 175–185.

[20] A. Gelman, J. Hill, Data Analysis Using Regression and Multilevel/Hierarchical Models, Analytical Methods for Social Research, Cambridge University Press, 2007.

[21] L. De Cicco, S. Mascolo, An adaptive video streaming control system: modeling, validation, and performance evaluation, IEEE/ACM Trans. Netw. 22 (2) (2014) 526–539.

[22] L. De Cicco, V. Caldaralo, V. Palmisano, S. Mascolo, Elastic: a client-side controller for dynamic adaptive streaming over http (dash), in: Proceedings of the Packet Video Workshop, 2013, pp. 1–8.

[23] G. Cofano, L. De Cicco, S. Mascolo, Characterizing adaptive video streaming control systems, in: Proceedings of the American Control Conference, 2015, pp. 2729–2734, doi:10.1109/ACC.2015.7171147.

Luca De Cicco received the computer science engineering degree (Hons.) and the Ph.D. degree in information engineering from the Polytechnic of Bari, Bari, Italy, in 2003 and 2008, respectively. Currently, he is an Assistant Professor at Politecnico di Bari. He has held visiting positions at the University of New Mexico, Albuquerque, NM, USA, in 2007; Ecole Superieure d'Electricité, Paris, France, in 2012; and the Laboratory of Information, Networking and Communication Sciences-LINCS, Paris, France, in 2013 and 2014. He is the co-author of more than 50 papers published in international journals, books, or conferences. His main interests are the modeling and design of congestion control algorithms for multimedia transport, adaptive video streaming, Software Defined Networks, and Server Overload Control.

Saverio Mascolo received the Laurea degree (Hons.) in Electronics Engineering and the Ph.D. both from Politecnico di Bari, Italy, in 1991 and 1994, respectively. Currently he is Full Professor and Chair of the Department of Electrical Engineering and Computer Science at Politecnico di Bari. He was a Postdoctoral Researcher in 1995 and a Visiting Researcher in 1999 at the University of California, Los Angeles (UCLA) and Visiting Consultant at the University of Uppsala, Sweden, from 2002 to 2004. He has authored or co-authored more than 120 papers in international journals, books, or conferences. He holds four U.S. and three Italian patents. He has worked on intelligent manufacturing systems, deadlock avoidance, nonlinear control, chaotic systems, synchronization of chaotic systems using observers, crypto communications using observers, modeling and control of data networks, congestion control, adaptive video streaming, content delivery networks, software-defined networks and server overload control. He has been Associate Editor of the IEEE Transactions on Automatic Control. Currently, he is Associate Editor of IEEE/ACM Transactions on Networking and of Computer Networks Journal, Elsevier. Since 2018, he is an IEEE fellow.

Vittorio Palmisano is a research fellow at Politecnico di Bari. He has graduated in Computer Science Engineering with honors in June 2006 and he received the Ph.D. in Information Engineering in April 2010 from Politecnico di Bari. His main research interests are related to network applications design and implementation, with a special focus on multimedia contents streaming over Internet and video conferencing systems.