Optimal Delivery of Rate-Adaptive Streams in Underprovisioned Networks

Jiayi Liu, Catherine Rosenberg, Gwendal Simon, and Géraldine Texier

***Abstract*—The growth of Internet video traffic imposes a severe capacity problem in today's Content Delivery Network (CDN). Rate-adaptive streaming technologies, such as the Dynamic Adaptive Streaming over HTTP (DASH) standard, reinforces this problem in the core CDN infrastructure since delivering one video means delivering multiple representations for an aggregated bit-rate that is commonly over 10 Mbps. In this paper, we explore better trade-offs between CDN infrastructure cost and Quality of Experience (QoE) of the end-users for live broadcast video streaming applications. We consider in particular underprovisioned CDN networks, our goal being to maximize the QoE for the population of heterogeneous end-users despite the lack of resources in the intermediate CDN equipments. We show that previous theoretical models based on elastic bit-rates do not fit for this context. We propose a user-centric discretized streaming model where the satisfaction of end-users is related to the context and where a stream has to be either delivered in its entirety, or not delivered at all. We first formulate an Integer Linear Program (ILP) that achieves the optimal delivery through a multi-tree delivery overlay. The evaluation of the ILP shows the benefits of this model. We then design a practical system by revisiting the three main algorithms implemented in CDN: user-to-server assignment, content placement and content delivery. At last, we use a realistic trace-driven large-scale simulator to study the performances of our system. In particular, we show that the population of users is reasonably well served (three quarters of the population do not experience degradation) even when the CDN infrastructure experiences a severe underprovisioning (less than half of the required infrastructure).**

***Index Terms*—rate-adaptive streaming ; user satisfaction ; underprovisioning ; optimization problem**

I. Introduction

In THE TRADITIONAL architecture for the delivery of live video streams in the Internet, we distinguish between the *content provider*, which originates the stream, the *Content Delivery Network* (CDN), which is in charge of delivering that stream, and the *end-users* (see Fig. 1). End-users form a heterogeneous population, which consumes video on a growing range of devices. The quality of their network connection to Internet can also differ a lot. To address this heterogenous population, content providers are adopting *rate-adaptive streaming* technologies (*e.g.* the standardized Dynamic Adaptive Streaming over HTTP (DASH) protocol). The main idea behind these delivery techniques is to enable CDN edge-servers to match video stream quality with any end-user, whatever her device/network configuration.

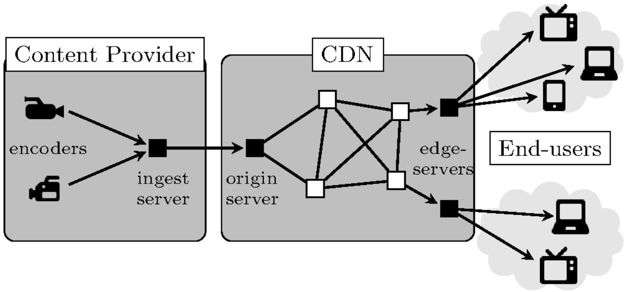


Fig. 1. Live stream delivery: the main actors

Rate-adaptive streaming is appealing for the delivery from edge-servers to end-users. It is however extremely demanding in the core CDN infrastructure. The high-level *channel stream* is indeed a collection of smaller *video streams*, each corresponding to a different *representation* of the video. For example, the Netflix HD videos are encoded into up to 14 representations with accumulated rate over 15 Mbps per channel stream [1]. The consequence is that CDN providers meet a scalability issue in their infrastructure [2].

The problem of scalability is more challenging for live streaming systems than for Video on Demand, for which it is possible to delay bulk transfers at off-peak hours [3]. Live streaming requires instead the CDN provider to *provision* a delivery infrastructure in advance, *i.e.* to make sure that equipments in the CDN infrastructure are able to transmit streams from origin servers to edge-servers. In the current paper, we consider the problem of delivering live rate adaptive streams within the CDN network. Although the DASH standard works in a pull way (between DASH clients and CDN edge servers), CDN providers should proactively push the live content to edge servers through network in order to serve the content to users in-time. Previous works in the area [4–6] have highlighted the limited outbound rate of intermediate equipments. We also consider that the most critical resource to provision is the overall amount of data equipments can emit per unit of time.

We consider the scenario where the CDN infrastructure is *underprovisioned*, which means it is not possible to convey all representations of requested contents to the edge-servers. The scientific interest for this topic is growing because CDNs look for better trade-offs between the infrastructure provisioning cost and the quality of service. In this paper, we explore the topic of streaming in underprovisioned network from an optimization standpoint. Our goal is to maximize the Quality of Experience (QoE) for the population of heterogeneous end-users despite the underprovisioning of CDN infrastructure.

Manuscript received April 15, 2013; revised September 22, 2013 and December 15, 2013.

J. Liu, G. simon, and G. Texier are with Télécom Bretagne, France (e-mail: {firstname.lastname}@telecom-bretagne.eu).

C. Rosenberg is with University of Waterloo Canada, (e-mail: cath@uwaterloo.ca).

Digital Object Identifier 10.1109/JSAC.2014.1404xx.

Previous theoretical works related to live streaming in CDNs [4, 5, 7, 8] have targeted the reduction of delivery transmission cost. However, the development of peering agreements between CDN and network operators has reduced the importance of transmission cost in CDN (see for instance agreements between Netflix and network operators [9]). We chose to neglect the bandwidth cost, and to focus on delivery issues in the CDN infrastructure. This trend has been initiated in a series of recent theoretical works [10–13], which have introduced an optimization problem coined as *streaming capacity problem*. The goal is to determine the maximum bit-rate that can be delivered from one source to a set of receivers when the infrastructure is made of nodes with limited data emission capabilities. These works assume *elastic* video bit-rate and infinitely divisible data streams. However, in the context of live rate-adaptive streaming, the video bit-rate of representations are pre-defined. Each stream has to be either delivered in its entirety, or not delivered at all. Moreover, some representations are more popular than others. We will show in this paper that the current formulation of streaming capacity problem does not satisfactorily address the problem met by underprovisioned CDNs with rate-adaptive streaming.

In this paper we propose a *discretized user-centric streaming* model, which is especially designed for live rate-adaptive streaming in modern CDNs. We formulate an optimization problem, which differs from the streaming capacity problem in two critical ways:

- Instead of maximizing the overall bit-rate, the goal is to maximize the number of streams that are delivered at the edge-servers, subject to the capacity constraints of the intermediate equipments in the CDN core network.
- Instead of serving all representations of a given channel to edge-servers requesting this channel, we prioritize the delivery of representations to the edge-servers according to the characteristics of the different populations of end-users.

Our first contribution is the formulation of an optimization problem using the user-centric discretized streaming model for a static context. It could be regarded as a snapshot of a dynamic system, and provides fundamental results for designing a practical system in dynamic environment. Our objective is first to maximize the minimum satisfaction among users, and then to maximize the average satisfaction of users. We formulate an Integer Linear Program (ILP), which jointly decides: (1) the representations that should be sent to the edge-servers, (2) the building of a delivery overlay, *i.e.* a set of delivering trees from origin servers to edge-servers, and (3) the assignment of end-users to edge-servers. The evaluation of the ILP model on a toy-CDN infrastructure highlights the interests for our model.

Our second contribution is the presentation of a practical implementation, which enables a CDN provider to efficiently deliver rate-adaptive live streams in dynamic environment. We propose algorithms for the three main processes of a CDN: *content placement*, *content delivery* and *user assignment*. We evaluate the performances of our system through a large simulation campaign. We utilized real traces from the justin.tv platform [14] to set up our simulator, which emulated up to $80,000$ simultaneous end-users. The results show that the system achieves high user satisfaction with limited CDN infrastructure in a dynamic environment.

The rest of the paper is organized as follows. We describe in §II the discretized user-centric streaming model. Our optimization problem is formally given in §III and quickly evaluated in §IV. We present in §V a practical implementation of a CDN system for delivering live rate-adaptive streams. We provide in §VI a first evaluation based on a large-scale real-trace simulator. §VII summarizes related work and §VIII concludes the paper.

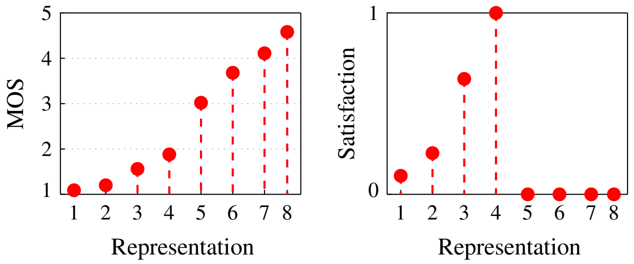
II. Discretized User-centric Streaming Model

When its delivery system is underprovisioned, a service provider cannot offer as many video representations as it would with a well-provisioned delivery system. As a consequence, the representation that fits the best a given user's downloading capability might not be offered by the CDN. This situation forces users to get a lower bit-rate video representation, hence with a lower quality. In the following, we introduce the concept of *user satisfaction* for every user and every representation. We propose a model that measures how far is the received representation to the best possible one for a given user. We detail this concept in §II-A. We define in §II-B our model for the delivery of multiple live video streams from the origin-servers to edge-servers.

A. User satisfaction

The ideal way to measure the QoE perceived by human viewers is to run subjective tests and to combine the obtained scores into a Mean Opinion Score (MOS). An MOS traditionally ranges from 1 to 5, where 1 stands for bad quality and 5 for excellent quality. However, subjective test campaigns are costly to conduct and require time. Researchers have thus worked on QoE models that map objective non-perceptual video quality metrics into MOS values [15–17]. The idea is that the combination of several paramaters (such as video bit-rate, packet loss, and video genre) can lead to a good estimation of the MOS. In particular, models based solely on video bit-rates represent a nice trade-off between the accuracy of QoE estimation and the simplicity of implementations [18–20]. In the following, we consider that the CDN provider uses such bit-rate based QoE model to estimate on-the-fly the MOS of a given video representation. One of the advantages of these models is that they do not require any computation on the user devices. Of course, more sophisticated QoE models, based on a wider set of parameters, can be implemented to improve MOS estimation accuracy.

We focus on the bit-rate parameters to measure QoE because it is a central parameter of rate-adaptive streaming. Indeed, (i) representations are encoded with pre-defined bit-rates, and (ii) each end-user is characterized by her highest viewable representation, which is the representation that the user can download with the highest bit-rate. In Fig. 2(a), we depict a typical *MOS profile* for a video. Without loss of generality, we combined the bit-rate QoE model given in [19] and the bit-rates recommended by Apple for the set of video representations in HTTP Live Streaming (see Table III



(a) Mean Opinion Score for a channel with eight representations.

(b) Relative satisfaction for users who can only play up to the fourth representation

Fig. 2. Introducing a user-centric measure of satisfaction

and [21]). As can be expected, the MOS grows when the encoding bit-rate increases, but the growth is not linear.

However, such MOS-based models of representations do not fully capture the satisfaction of a heterogeneous population of end-users. Indeed, the QoE depends on the context of video consumption [22]. For example an end-user watching a 320p video on a smartphone in public transportation is satisfied although the same video is perceived as being of very bad quality by the same end-user watching it on her TV screen. Furthermore, various reasons can prevent an end-user to watch high-quality video representations, including too small screen resolution and not enough bandwidth, hence receiving a 720p video may lead to the worst user satisfaction as such video cannot be played at all.

Previous context-aware QoE models such as [22] do not consider multi-representation streaming. In this paper, we introduce the *relative satisfaction* metric (or satisfaction in short). We normalize the satisfaction of a given user getting a given representation to the satisfaction obtained if she gets her best possible representation. This relative satisfaction is thus between 0 and 1. We show in Fig. 2(b) the relative satisfaction of an end-user who cannot play a video above the fourth representation. We used the same MOS as in Fig. 2(a). Please observe in Fig. 2(b) that the third representation has a good relative satisfaction although the same representation has a disastrous MOS in Fig. 2(a).

Let us now introduce some notations. The set of end-users is noted N. The live streams consist of l different channels. The raw video of each channel is transcoded into k representations, where the bit-rate of the i-th representation, $1 \leq i \leq k$, is λ_i. For simplicity of notation, hereafter we denote by $[m]$ the integer set $\{1, \ldots, m\}$. Also, let d_{ij} be the i-th representation of the j-th channel, $i \in [k]$, $j \in [l]$.

For each user $n \in N$, we denote by i^n the index of the best possible representation, and by j^n the index of the channel she is watching. Thus, $d_{i^n j^n}$ is the best possible representation requested by user n. We then define a function $MOS(d_{ij}, n)$, which is the QoE rating of user n for representation d_{ij}. In comparison with the plain MOS profile (Fig. 2(a)), this MOS score is user-related. There are two cases:

- End-user n can decode and watch d_{ij} ($i \leq i^n$ and $j = j^n$). In this case, the value of $MOS(d_{ij}, n)$ is the plain MOS profile of the representation $MOS(d_{ij})$.
- End-user n cannot decode and watch d_{ij} ($i > i^n$ or $j \neq j^n$). In this case, we "force" $MOS(d_{ij}, n)$ to be equal to one (the least MOS score value). In other words, an end-user is totally unsatisfied when getting an unwatchable representation.

We note u_{ij}^n the satisfaction of user $n \in N$ when n receives representation d_{ij}. To compute u_{ij}^n, we use the ratio of the MOS of representation d_{ij} to the MOS of its requested representation $d_{i^n j^n}$. Since we would like the user satisfaction to be into the range $[0, 1]$, we use the following computation:[1]

$$u_{ij}^n = \frac{MOS(d_{ij}, n) - 1}{MOS(d_{i^n j^n}, n) - 1} \tag{1}$$

B. Live video streaming in a CDN

The topology of a CDN is modeled by a directed graph $G = (V, E)$, where V represents the CDN devices, and E represents the communication links. Previous works on live streaming in CDNs [5, 8] have shown that major CDNs now have three types of equipments (nodes): a relatively small number of *sources* (origins), a medium size network of *reflectors*, and a large number of *edge-servers*. The sources receive and transcode the raw video channels into a set of live representations; the reflectors deliver the representations to the CDN edges, and the edge-servers offer the received representations to the end-users. Let $V_S, V_R, V_E \subset V$ be the set of sources, reflectors and edge-servers, respectively.

The delivery of a representation d_{ij}, $i \in [k]$, $j \in [l]$, from the source nodes to the edge-servers is carried out through a set of subtrees of G. Each tree, also referred to as the *delivery tree*, has one of the source nodes as its root and edge-servers as its leafs. We denote by T_{ij}^s the delivery tree of d_{ij} rooted at $s \in V_S$.

Note that every *forwarding node* v, either a source or a reflector, is limited by the total outbound bit-rate (capacity) it can support, $c(v)$. Like all previous works [4–6], the outbound capacity of the equipment is the only constraint. Furthermore, we neglect some low-level networking issues, such as bandwidth fluctuation generated by on-off traffic [23]. In its inner network, a CDN provider can apply global traffic management techniques to mitigate these issues.

III. Formulation of the Capacity Problem

We start the presentation of the optimization problem by introducing several objective functions in §III-A. We then present the ILP formulation in §III-B.

A. Objective Functions

Ultimately we would like every edge-server to receive all the representations of the video channels that are requested by its end-users. This however might not be possible due to the outbound capacity constraints at the forwarding nodes. In that case, the CDN supports the delivery of a subset of representations for each edge-server. To formalize the discretized user-centric streaming capacity problem, we define two binary variables:

[1] We assume that the plain $MOS(d_{ij})$ is larger than 1 for all representations, because logarithmic QoE models are used [19].

- $x^e_{ij} = 1$ indicates that edge-server $e \in V_E$ receives representation d_{ij}, 0 otherwise.
- $z^{ne}_{ij} = 1$ indicates that user $n \in N$ is attached to edge-server $e \in V_E$, and gets representation d_{ij} from e, 0 otherwise.

We suppose that the highest achievable representation for each user n is given. It is determined by user device type and access link. As mentioned in II-A, the user satisfaction u^n_{ij} is known for all users and all representations.

It is possible to design many different objective functions. We propose one user-centric objective function in the following. An intuitive objective function consists in maximizing the average satisfaction of users, which is expressed as follows:

$$O_{global} : \sum_{e\in V_E} \sum_{i\in[k]} \sum_{j\in[l]} \sum_{n\in N} u^n_{ij} \cdot z^{ne}_{ij} \tag{2}$$

Such an objective function is simple, but it has some weaknesses. In particular, it sometimes introduces biases, when O_{global} can be maximized by focusing on a subset of the population. It is the case here: users watching low-resolution representations will be served first because satisfying them has a lower impact on the infrastructure in comparison to satisfying users watching high-definition channels. We thus need another objective function, which avoid such bias. A way to mitigate such weaknesses is to ensure some degree of fairness among the end-users. A simple user-centric objective is formulated as a *max-min*, where the goal is to maximize the satisfaction of the user that is the most poorly served, which is formally given by:

$$O_{maxmin} : \min \left(\sum_{e\in V_E} \sum_{i\in[k]} \sum_{j\in[l]} u^n_{ij} \cdot z^{ne}_{ij} | n \in N \right) \tag{3}$$

In this paper, we decide to put emphasis on the max-min objective depicted in Eq.(3) due to the aforementioned weaknesses. Therefore, we define the overall objective function as:

$$\textbf{maximize} \quad \varepsilon \cdot O_{global} + O_{maxmin} \tag{4}$$

where the parameter ε is chosen small enough to guarantee that O_{maxmin} always dominates O_{global}. In other words, with this formulation, we first aim to ensure that all end-users have a decent satisfaction, and then we maximize globally the satisfaction of the whole population. A simple way to set the value of ε is as follows. Firstly we observe that O_{maxmin} is a sequence of discrete values. We want O_{global} to be always smaller than the "interval" between two consecutive user satisfactions so that O_{maxmin} always prevails. The smallest interval between two consecutive user satisfactions is $\min\{u^n_{ij} - u^n_{i'j} | \forall n, i \neq i'\}$. Since the objective O_{global} cannot be greater than $|N|$, we can set $\varepsilon < \frac{1}{|N|} \cdot \min\{u^n_{ij} - u^n_{i'j} | \forall n, i \neq i'\}$. Please note that other objective functions can also be considered. For example an interesting objective would be to maximize the number of users who receive their best representation: $\sum_{e\in V_E} \sum_{i\in[k]} \sum_{j\in[l]} \sum_{n\in N} z^{ne}_{i^n j^n}$.

Overall, the *user-centric discretized streaming capacity problem* is defined as: given the topology and capacity constraints of a CDN, find the delivery tree sets, $\{T^s_{ij}\}_{i\in[k],j\in[l]}$, such that the user-centric objective (4) is maximized.

Linear Programming formulation

$$\textbf{max.} \quad \varepsilon \cdot O_{global} + O_{maxmin}$$

$$O_{global} : \sum_{e\in V_E} \sum_{i} \sum_{j} \sum_{n\in N} u^n_{ij} \cdot z^{ne}_{ij}$$

$$O_{maxmin} : \min(\sum_{e\in V_E} \sum_{i} \sum_{j} u^n_{ij} \cdot z^{ne}_{ij} | n \in N)$$

$$\begin{aligned}
\textbf{s.t.} \quad & \sum_{e\in V_E} \sum_{i} \sum_{j} z^{ne}_{ij} \leq 1, && \forall n && (5)\\
& z^{ne}_{ij} \leq x^e_{ij}, && \forall n, e, i, j && (6)\\
& z^{ne}_{ij} \leq p^e_n, && \forall n, e && (7)\\
& z^{ne}_{ij} < u^n_{ij} + 1, && \forall n, e, i, j && (8)\\
& \sum_{n\in N} \sum_{i} \sum_{j} z^{ne}_{ij} \cdot \lambda_i \leq c(e), && \forall e && (9)\\
& x^e_{ij} \leq \sum_{s\in V_S} I^u_{ijs}(V_R) && \forall i, j, e && (10)\\
& I^r_{ijs}(V_S \cup V_R) \leq 1 && \forall i, j, s, r && (11)\\
& I^e_{ijs}(V_R) \leq 1 && \forall i, j, s, e && (12)\\
& \sum_{i=1}^{k} \sum_{j=1}^{l} O^s_{ijs}(V_R)\lambda_i \leq c(s) && \forall s && (13)\\
& \sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{s\in V_S} O^r_{ijs}(V_R \cup V_E)\lambda_i \leq c(r) && \forall r && (14)\\
& h^s_{ijs} = 0 && \forall i, j, s && (15)\\
& h^r_{ijs} + 1 - h^v_{ijs} \leq |V|(1 - y^{rv}_{ijs}) && \forall i, j, r, v && (16)\\
& O^r_{ijs}(V_R \cup V_E) \leq |V|(I^r_{ijs}(\{s\} \cup V_R)) && \forall i, j, s, r && (17)\\
& I^r_{ijs}(\{s\} \cup V_R) \leq O^r_{ijs}(V_R \cup V_E) && \forall i, j, s, r && (18)
\end{aligned}$$

B. Integer Linear Program Formulation

We formulate an ILP model for the user-centric discretized streaming capacity problem. Besides both binary variables defined in §III, we define a new binary variable for tree construction:

- $y^{uv}_{ijs} = 1$ indicates that edge $(uv) \in E$ is used in the tree T^s_{ij}, 0 otherwise.

In CDN, it is frequent that an edge-server is assigned to a given population of end-users, or a geographic area. For example, an edge-server can be located within the network of an Internet Service Provider (ISP) to serve exclusively clients of this ISP. We define another binary variable:

- $p^e_n = 1$ indicates that user $n \in N$ can be assigned to edge-server $e \in V_E$, 0 otherwise.

To ease notations, we define $I^v_{ijs}(U)$ as the sum of the y variables that correspond to the incoming edges into $v \in V$ from the nodes in $U \subseteq V$. Similarly, let $O^v_{ijs}(U)$ be the sum of the y variables that correspond to the outgoing edges from $v \in V$ to nodes in $U \subseteq V$. For every node $v \in V$, h^v_{ijs} is an upper bound on the depth of v in T^s_{ij}.

Constraint (5) makes sure that users get at most one representation from one edge-server. With constraint (6) edge-server send only the representations they receive. Constraint (7) redirects users to proper edge-servers. Constraint (8) does not allow the delivery of representations that are not

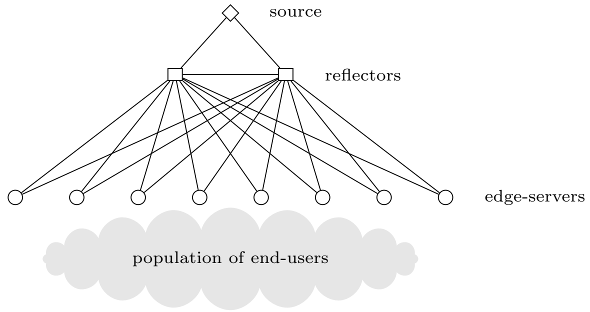


Fig. 3. Topology of CDN toy-network

associated with a positive satisfaction. Constraint (9) restricts edge-server capacity. Then, the remaining constraints construct trees from source to edge-servers. Constraint (10) indicates that if there is no incoming edge, the stream is not received on the edge-server. Constraints (11) and (12) enforce a single parent in every tree. Constraints (13) and (14) enforce capacity restrictions. Constraints (15) and (16) prevent cycles in every tree. Constraint (17) states that if there is no incoming edge in a tree, there cannot be outgoing edges as well. Finally Constraint (18) forces reflectors to have at least one output edge if they have an incoming edge in a tree.

This ILP model solves the three main CDN functionalities all at once. The optimal solution jointly indicates : (1) which representations are sent to which edge-servers; (2) how to deliver streams from CDN sources to CDN edge-servers; and (3) how users are assigned to edge-servers.

IV. Proof-of-Concept for Discretized Streaming

We now show the relevance of our approach to address the challenge of maintaining a high satisfaction on a population of users in a context of an underprovisioned CDN. We implemented the ILP model in IBM ILOG CPLEX optimizer and computed the best achievable delivery on a *toy CDN network*. We compare this optimal delivery (called *our approach*) with a solution that can be directly obtained from the streaming capacity problem. We computed the overall streaming capacity of the toy-network by implementing the algorithm described in [10]. Then, we found the best packing of representations for the computed capacity. We refer to this approach as the *traditional approach*. We show hereafter that the traditional approach does not provide an accurate computation of the optimal video delivery in the context of adaptive streaming. This justifies our new model, and the need of new algorithms based on this model.

We consider a small static toy-network, which represents a CDN delivering only one channel of 8 representations to a population of end-users. This toy-network consists of one source, two reflectors and eight edge-servers (as shown in Fig. 3). The bit-rates of the representations are set according to the recommended Apple HTTP Live Streaming [21]. We considered a population of 360 end-users interested in this channel. The number of users is set so that each representation is requested, thus the result obtained can reasonably reflect a real configuration. The characteristics of devices and network connections (see Table II) are inspired by some recent statistics collected during London Olympic games [24].

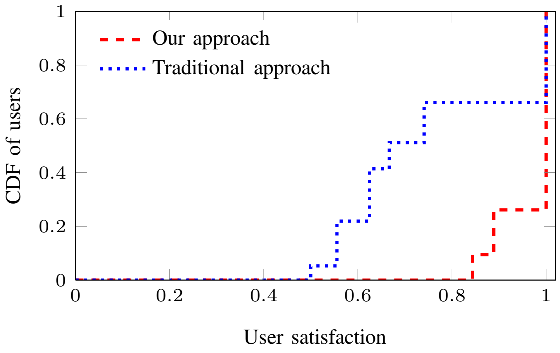


Fig. 4. CDF of user satisfaction

We assumed a scenario where the CDN provider has severely underprovisioned its infrastructure. The upload capacity of each source is 10 Mbps, which is enough to send all representations. However, the upload capacity of each reflector is only 7.2 Mbps, which is insufficient. The aggregated upload capacities of reflectors represent only 16% of what would be required to send all representations to all edge-servers. Finally, the capacity of edge-servers is 80 Mbps. There is no constraint on the assignment from end-users to edge-servers.

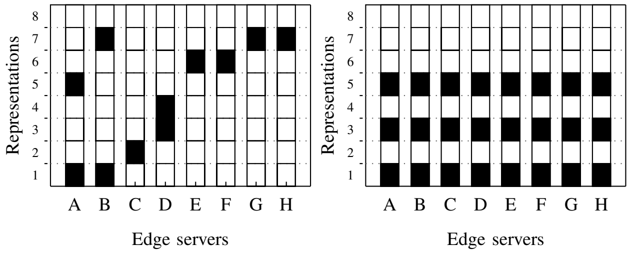
Our goal here is not to mimic the reality (we conducted realistic simulations in § VI). It is rather a proof-of-concept to illustrate the benefits one can expect from our model. The toy-network is small because, even for such a small network, the computation time is significant: it took us an average of about $5,000$ seconds per run to get the optimum solution on a regular server (2×4 cores Intel(R) Xeon(R) 2.67GHz CPUs). Furthermore, we considered a static environment, which is a snapshot of a real system (we will explore dynamic environments in the next sections of the paper). Note that the algorithms in [10] also consider a static network when they compute the network capacity.

To measure performances, we compute the Cumulative Distribution Function (CDF) of the satisfaction of users (see Fig. 4). A point at $(0.6, 0.2)$ means that 60% of users experience a relative satisfaction lower than 0.2.

We can immediately see in Fig. 4 the weaknesses of the traditional approach based on the streaming capacity problem. The QoE is far from what can be obtained with a dedicated approach. With the traditional approach, two thirds of users have a satisfaction below 0.75 although there exist solutions where all users have a satisfaction over 0.82.

To better understand the poor performance of the traditional approach, we plot in Fig. 5 the received representations on each edge-server. A black square indicates the representation is received. Please note that this simulation is for a static environment, thus, representation adaptation is omitted here. In our approach, all representations (except the highest one) are hosted in at least one edge-server. On the contrary, the traditional approach provides the same set of representations to all edge-servers. Our approach provides a solution with higher representation diversity, thus, more users can obtain their required representation.

Moreover, the ILP is able to adjust the number of representations to both the capacity of the edge-servers and the demand



(a) Our approach (b) Traditional approach

Fig. 5. Received representations on each edge server

from end-users. For example, edge server B receives two extremely different representations (the lowest representation 1 and the highest receivable representation 7) but most of its capacity is utilized to deliver the high representation (it serves representation 7 to 32 users, and representation 1 to only 2 users). It means that only three servers (servers A, C and D) are able to serve almost all users requesting the four lowest representations. This calls for specializing edge-servers on representations that are mainly accessed by *a class of users*.

Our second important observation is that our optimization model is able to find solutions with excellent performances. This is especially visible in Table I. Almost three quarters of the population do not experience any degradation although the CDN capacity is less than one fifth of what it should be. Based on these promising results for the static environment, we developed algorithms that can be implemented in a real system by taking into account system dynamics. In the following, we describe these algorithms and a performance evaluation.

V. A Practical System: SCADOOSH

We describe now a practical implementation for CDNs. The overall system is named SCADOOSH, which stands for SCAle Down fOOtprint for live daSH. SCADOOSH aims to efficiently exploit CDN infrastructure to deliver live video streams to a dynamic, large population of users. In the following, we use the nomenclature of the DASH standard, but SCADOOSH is independent of the implemented rate-adaptive technology.

In addition to the traditional 3-tier CDN infrastructure, SCADOOSH requires a centralized organizer, which we call a *coordinator*. The coordinator is in charge of orchestrating the delivery into a CDN network. Most CDNs rely on such a global coordinator, which manage the whole network or a restricted area. A SCADOOSH coordinator executes three main algorithms:

- *User assignment* algorithm aims to assemble users with similar requirements. The result of this algorithm is a user distribution over the edge-servers.
- *Content placement* algorithm calculates the *utility* of each stream for each edge-server which measures how users on the edge-server will be satisfied with the stream. The utility is based on the reports from edge-servers about the activity of the served population.
- *Delivery trees construction* algorithm determines a utility-driven multi-tree overlay that delivers videos from sources to edge-servers.

TABLE I

Number of viewers per video degradation level, $deg - x$ means a user getting a representation that is x levels below her best possible representation

	Our approach	Traditional approach
deg-0	266	122
deg-1	94	108
deg-2	0	70
deg-3	0	60

The stream is cut into time *periods* such that, at the end of every period, users update their Media Presentation Description (MPD) file. The implementation of periods can typically be done through the field `minimumUpdatePeriodMPD` in the DASH standard. Then, SCADOOSH synchronizes on these periods to cope with the dynamic feature of rate-adaptive live streaming. At the end of every period, edge-servers report to coordinator about the popularity of each representation of each channel during the last period (the number of users requesting each representation on edge-servers). Upon reception of these reports, the coordinator estimates the popularity of representations for the next period. This popularity forecast enables the computation of stream utilities. Then, the coordinator determines a multi-tree delivery overlay that maximizes the sum of utility scores. During the whole next period, sources and reflectors use the multi-tree overlay to deliver the live streams. The edge-servers are informed about the streams that they will receive during the next period, and build their MPD files accordingly. Then, users update the MPD file and use it to fetch upcoming content for the next period.

A. Type Specified User Assignment

How to assign an end-user to an edge-server when the end-user starts a streaming session is generally the responsibility of the CDN providers through a DNS redirection scheme. A tool like GeoDNS has been used for years on that purpose [25]. In the case of proprietary delivery platforms, studies showed that the redirection depends on both network proximity, load-balancing [26] and business issues [27]. In conformance with [28], we assign here every end-user to only one edge-server and *we promote a policy where the end-users are assigned to edge-servers according to the characteristics of their devices and network connections*. Some network scientists have advocated such policy before us [29]. We will show later that significant gains in terms of CDN infrastructure can be achieved if the CDN is able to adequately redirect requests with regard to the specialization of edge-servers. In the following, we present a practical (*a.k.a.* simple) implementation of such *type-based user assignment*.

The principle is as follows. We roughly distinguish three families of streaming sessions: `mobi`, `HD` and `norm`. In short, the `mobi` family is for users with mobile devices and low bandwidth network connections, the `HD` for users with high profile, and last, the `norm` is the non-specialized family, which stands for other types of users. Note that finer divisions with a higher number of families can be possible. End-users naturally drive the choice of the session family: an end-user watching the channel from her smartphone or having a poor network connection falls naturally in the `mobi` family while an end-user consuming the video on a High-Definition TV is a `HD`.

With rate-adaptive streaming technologies, each family can be associated with a subset of representations. Our goal is to specialize edge-servers so that each edge-server is also associated with a family. Indeed, we would like to leverage the fact that HD representations are rarely demanded by edge-servers in the `mobi` family while `HD` edge-servers should not require low-quality representations.

The technologies to identify users' family is out of the scope of this paper. Mobile device detection technologies have been developed from the earliest days of the mobile web. Typically the HTTP "User-Agent" header field allows any server to distinguish HTTP requests from mobile devices. Other quick testing can be done at session openings to roughly test the network connection. Here, we assume that the coordinator is able to identify the family of every user.

The coordinator is in charge of setting the family for each edge-server. Strategies to dynamically change the configuration of edge-servers according to users demand have been recently developed [30]. Such work complements our proposal.

Once a user enters the system, the user assignment algorithm is as follows:

1) The algorithm first searches for edge-servers (1) from the same family, (2) within the same *area* (*e.g.* ISP), (3) hosting users watching the same channel, and (4) with load lower than a pre-defined threshold. If some edge-servers match these requirements, the user is assigned to one of them by a random choice.
2) Otherwise, the algorithm restricts lookup to `norm` edge servers (1) within the same area, (2) hosting users watching the same channel, and (3) with load lower than the threshold. Similarly, the user is assigned to one of such servers.
3) Finally, if no edge-server is found in the above steps, the user is assigned to the `norm` edge-server with the minimum traffic load among edge-servers within the same area and with load lower than the threshold. Otherwise, the system cannot provide live video service for this user.

Then, the user fetches a MPD file from the edge-server, which states the base URL of the video, the available video representations, segments and their relative URLs. After parsing the MPD file, the user requests the upcoming segments and plays the video.

B. Utility-based Content Placement

The intuition behind content placement is that we need to prioritize the delivery of certain representations over others for each edge-server. We introduce the concept of *utility at the edge-server* to enable such prioritization. The utility is a generic concept. In this paper, we propose a simple function, where the utility of a representation d_{ij} at edge-server $e \in V_E$, denoted by α^e_{ij}, is expressed as the potential aggregated satisfaction of all users that are served by e. We compute the utility of a representation at an edge-server by counting the number of requests for this representation. Let us denote by n^e_{ij} the number of user requests for representation d_{ij} at edge-server $e \in V_E$. Since a user n is unsatisfied ($u^n_{ij} = 0$) for a representation that is either from a different channel or higher than its requested one, we compute the utility α^e_{ij} as:

$$\alpha^e_{ij} = \sum_{i' \geq i} \sum_{j'=j} n^e_{i'j'} \frac{MOS(d_{ij}) - 1}{MOS(d_{i'j'}) - 1} \qquad \forall i, j, e \qquad (19)$$

A high utility for a representation d_{ij} for e means that a large number of end-users served by e can be highly satisfied by d_{ij}. Therefore the representation d_{ij} should be delivered to e in priority. To cope with system dynamics, the coordinator should know the requests that will be issued *during the next period*. The main challenge is that the number of requests for a given representation at an edge-server can dramatically and unpredictably change between two consecutive periods. Indeed, rate-adaptive streaming systems are dynamic and controlled by end-users. To address this problem, we propose in the following a solution based on *time series forecasting*, where the coordinator leverages requests from previous periods to predict the requests for the next period, for every representation on every edge-server.

The forecasting model that we utilized is called Autoregressive Integrated Moving Average (ARIMA) [31]. This model has already proven its efficiency for the popularity of channels in IPTV system [32]. The algorithm is as follows:

1) Until sufficient records are collected (50 periods are suggested in [31]), the coordinator anticipates that the number of requests for each representation at each edge-server in the next period will be exactly the same as in the previous period.
2) When enough records have been collected, the parameters of the identified ARIMA model are estimated ; predictions can be made more accurately.
3) Previous predictions are confronted to real requests. If the performances are below a threshold, the ARIMA parameters should be retrained.

More formally, let T be the current period, so the coordinator should predict the requests for period $T+1$. If reports have been sent from edge-servers to coordinator, the latter stores the times series $n^e_{ij}(t), \forall i \in [K], j \in [L], e \in V_E$ for all $t \leq T$. For simplicity, we assume that all these time series are independent, thus we treat them separately.[2] Based on the ARIMA model, the prediction $\widehat{n^e_{ij}}(T+1), \forall i \in [K], j \in [L], e \in V_E$ can be calculated as follows:

$$\widehat{n^e_{ij}}(T+1) = 2n^e_{ij}(T) - n^e_{ij}(T-1) + a^e_{ij}(T+1) - \theta^e_{ij} a^e_{ij}(T) \qquad (20)$$

where $n^e_{ij}(T)$ and $n^e_{ij}(T-1)$ are the records in periods T and $T-1$, and a^e_{ij} stands for random errors. In forecasting, the error for the future is treated as zero, thus $a^e_{ij}(T+1) = 0$. The error for the past is estimated as the difference between the real value and the predicted value, thus $a^e_{ij}(T) = n^e_{ij}(T) - \widehat{n^e_{ij}}(T)$. The coefficient θ^e_{ij} can be estimated from the least squares algorithm. To cope with dynamicity, we used the method suggested in [32]: For each prediction $\widehat{n^e_{ij}}(T)$, we compare the real value $n^e_{ij}(T)$ against its 95% confidence interval. If five continuous real values lie outside the interval, the latest

[2] This is not true in rate-adaptive streaming. We left for future works the design of more accurate prediction tools, which take into account the correlations between various representations.

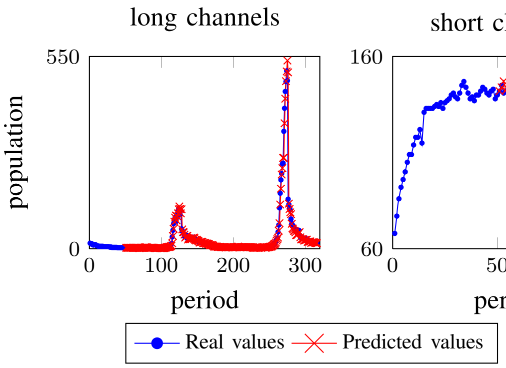


Fig. 6. Evaluation of the prediction, based on ARIMA

fifty observations are used to retrain the θ_{ij}^e parameter, and the new θ_{ij}^e is used for the following prediction.

To validate our model, we collected video popularity data from the justin.tv platform [14]. In comparison to IPTV channels, we distinguish two types of user-generated channels: long (respectively short) channels with (respectively without) peaks. We show the quality of the video population prediction for both video types in Fig. 6. We utilize the standard way to show the performances of such prediction tools with overlapping curves. We conclude that the ARIMA model is capable to accurately predict population changes.

At last, the predicted value obtained from Eq. (20) can be substituted in Eq. (19) to obtain the utility of the representations for the next period.

C. Utility driven delivery trees construction

We describe now the algorithm that builds a multi-tree delivery overlay over the CDN infrastructure. The main idea is to create delivery links between an edge-server and one of the equipments (source or reflector) that are able to deliver the representation. We process representations iteratively based on the *utility score per rate unit* (uspru) which is defined as $\frac{\alpha_{ij}^e}{\lambda_i}, \forall i, j, e$. The CDN provider first sorts the set of usprus (line 1 in Algorithm 1), then it processes each representation iteratively (lines 3-7).

Algorithm 1: Delivery forest overlay construction

```
1: $\mathcal{L} \leftarrow$ pile $\{e, d_{ij}\}$ by decreasing uspru order
2: transform graph $G$ into graph $G'$
3: while $\mathcal{L} \neq \emptyset$ and $V' \cap (V_S \cup V_R) \neq \emptyset$ do
4:     $e, d \leftarrow$ pop $\mathcal{L}$
5:     $G'_d \leftarrow$ augment $G'$ with node $t$, edges $(t, s)$, $s \in V_S$
       and edges $(t, r)$, $r \in V_R$ receiving $d$
6:     find max-residual path between $e$ and $t$ in $G'_d$
7:     update $G'$ with new capacities
```

Let e be the edge-server that has to be served. Let d be the representation that has to be delivered, with bit-rate λ. We aim at delivering representation d to edge-server e *while minimizing the impact on the infrastructure*. Our algorithm is inspired by the Maximum Residual Energy Routing Path algorithms, which are commonly used in wireless sensor networks with the goal to prolong system lifetime [33]. Instead of picking the shortest path between e and a device having d, which can quickly drain some equipments and make the network partitioned, we look for the delivery path such that the minimum remaining available upload capacity of all equipments in the path is maximum.

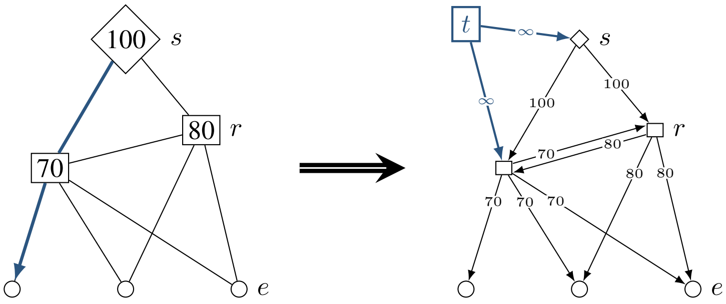


Fig. 7. Graph transformation

To achieve this goal, we transform the original node capacitated infrastructure graph G to an edge capacitated one G'_d. The transformation is illustrated in Figure 7. Then, delivering a representation d can be regarded as sending a flow of λ to e, and the delivery link corresponds to the maximum residual capacity path in the transformed graph G'_d. The transformation is in two steps:

1) G is first transformed to a directed graph by changing the original non-directed links to directed ones. We replace a link between two reflectors (u, v) in V_R by two links $(u, v, c(u))$ and $(v, u, c(v))$ with each link weighted by the capacity of the head vertex. For links that connect sources to reflectors and reflectors to edge-servers, the original link is replaced by a directed one from sources to reflectors and from reflectors to edge-servers respectively, still weighted by the capacity of the head vertex. This is because the video streams are never sent in the reverse direction.
2) Then, the directed graph is further transformed to a graph G'_d to facilitate the discovery of the delivery path for representation d. We add an abstract node denoted by t. The node t is linked to every node which can serve the representation with an infinite link.

From graph G'_d, it is trivial to find the path from the edge server e to the abstract node t with maximum residual weight. Especially, traditional shortest path algorithms such as the Dijkstra's one can be applied. With a standard implementation based on Dijkstra's algorithm with ordinary lists, the time complexity of the trees delivery overlay construction algorithm is $O(k \cdot l \cdot |V_E| \cdot |V|^2)$. More specifically, the average computation time for such a basic implementation is 43 seconds on a regular server (2×4 cores Intel(R) Xeon(R) 2.67GHz CPUs) with our simulation setting (we introduce our simulation setting in § VI).

VI. Evaluation

We now evaluate the performances of SCADOOSH. We developed our own simulator in C++ with the objective of simulating dynamic and large-scale systems. The simulator

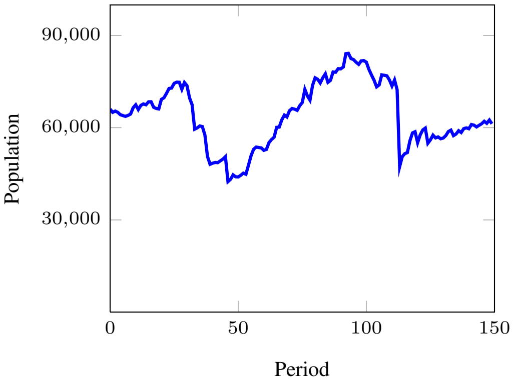


Fig. 8. Variation of global population over the 150 periods

TABLE II
TECHNOLOGIES AND RATIO OF ASSOCIATED USERS

Technologies	Average bandwidth (in Mbps)	Ratio of users
ADSL-slow	2	10%
ADSL-fast	8	30%
FTTH	100	10%
Wifi-slow	1	15%
Wifi-fast	5	15%
3G	0.8	20%

TABLE III
REPRESENTATION BIT-RATES

Representation	1	2	3	4
Bit-rate (in kbps)	150	240	440	640
Representation	5	6	7	8
Bit-rate (in kbps)	1,240	1,840	2,540	4,540

emulates a CDN system that delivers a number of live channels to a heterogenous population. The behavior of the population is inspired from the real traces obtained from *justin.tv*. Each user is characterized by a download bandwidth, which limits the representations she can get. Periodically, the system counts the utility of the representations and constructs delivery trees. Live streams are delivered through the trees. Then, we measure users satisfaction on the received videos under various delivery network provisioning conditions. We describe the settings of this simulator in §VI-A, and then we analyze the results in §VI-B.

A. *Simulation settings*

We fetched real traces from a popular user-generated live video broadcasting platform, named justin.tv [14], in August 2012.[3] Every five minutes, we retrieved the popularity of channels of justin.tv through the REST API provided by the platform. The 5-minutes interval corresponds to a typical MPD update period of the system. We selected this duration to balance the impact of system dynamics and the overhead caused by too frequent MPD updates. Also, this duration is long enough for the execution of all algorithms on our server. We took 150 of these measures to simulate the variation of popularity of channels in our system. We focused on the 50 most popular channels. We represent in Fig. 8 the global population of end-users in our simulations. At the peak level, the system hosts more than $80,000$ users.

We took inspiration from [24] to set the population heterogeneity. We defined six types of user network connections: ADSL-slow, ADSL-fast, FTTH, WIFI-slow, WIFI-fast and 3G. We show in Table II the average downlink bandwidth and the average ratio of the population. Along the simulation time, the downlink bandwidth of each user varies around the given average value. At each period starting time, users request the best representation they can get subject to their downlink speed.

We built a large CDN with five geographic *areas*. In each area, equipments are inter-connected through a *random graph* in which every link occurs independently with probability 0.8. We selected 30 among the 50 channels to be *globally hot*, which means that they are accessed by users from any area, while the remaining 20 are *locally hot* channels, they are viewed only from users in one area. The capacity of CDN equipments is set to 1 Gbps. In order to serve all users at peak time (no user is rejected due to overloaded edge-servers), the CDN contains 320 edge-servers.

To value the underprovisioning of the CDN infrastructure, we changed the number of reflectors. We set four underprovisioning configurations with respectively 15, 20, 25, and 30 reflectors. The former one corresponds to a *severely* underprovisioned infrastructure scenario where only 47 Mbps in average have been reserved per edge-server. Recall that, according to Apple HTTP Live Streaming setting [21] (see Table III), one channel is a pack of eight representations with an aggregated bit-rate over 11.6 Mbps. The most favorable scenario with 30 reflectors is a *slightly* underprovisioned infrastructure scenario.

B. *Results*

We now analyze the performances of SCADOOSH. Our main comparison is the ideal case where all users are served with their best representation. This comparison is captured by our metric based on relative satisfaction. In §VI-B1, we observe whether it is possible to maintain a good user satisfaction despite infrastructure underprovisioning. In §VI-B2, we focus on the impact of type-based user assignment on the overall performances.

1) Overall Performances: We use the same CDF function as in Fig. 4 to show the performances of SCADOOSH. We represent three curves, corresponding to three different number of reflectors. Results are given in Fig. 9

Our main observation is that SCADOOSH succeeds in maintaining a good QoE despite the lack of resources in CDN infrastructure. With 25 reflectors, only 6% of users are not served with their best representation. Moreover, most of them have a satisfaction over 0.88. In other words, the QoE is nearly perfect although the CDN slightly underprovisions its infrastructure.

Let us now highlight the severely underprovisioned scenario with only 15 reflectors. Again, SCADOOSH demonstrates its potential: 70% of users experience no degradation at all. Moreover, only 17% (respectively 9%) of users have a satisfaction below 0.88 (respectively 0.75). With respect to the severe underprovisioning of this infrastructure, these results

[3] Available at *http://enstb.org/∼gsimon/Resources/Justintv*

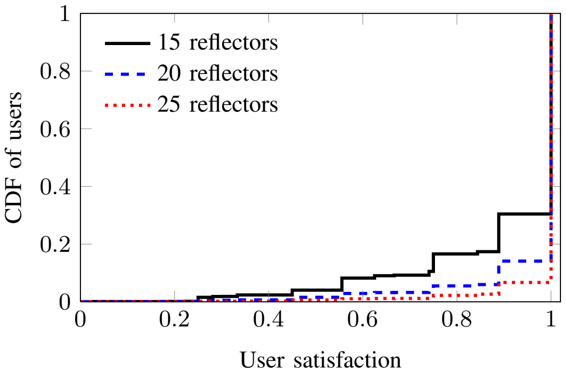


Fig. 9. CDF of user satisfaction for different underprovisioning

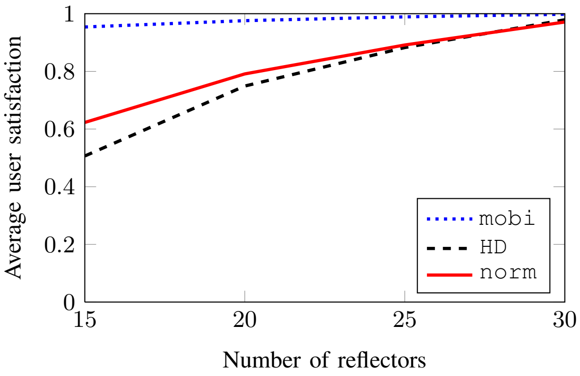
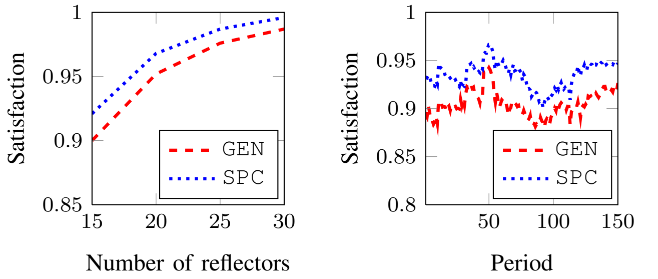


Fig. 10. Average user satisfaction for each edge-server type

are noteworthy since it shows that the population of users is reasonably well served although the CDN provisions less than half of the required infrastructure.

We would like now to get back to the optimal computation performed in the static configuration of § IV. Although both configurations differ a lot, typically because there are several channels in the dynamic simulations, we can compare the trends of both Fig. 4 and Fig. 9. In both cases, we observe a *long-tail* where the satisfaction of the most poorly served user is high. It means our practical algorithm succeeds in guarantying a good fairness among users.

Let us now have a closer look at the edge-servers from the same families in Fig. 10. As can be expected, end-users from the mobi family are almost not impacted by the underprovisioning. Indeed, the uspru of these clients is the highest, thus these users are prioritized in constructing the delivery trees. On the contrary, end-users from HD as well as norm are the ones that are the most affected by the underprovisioning. To complement these results, we investigate the video level distribution on the edge-servers. We calculate the average index of the representations that are available for each group of edge-servers with 15 reflectors. Higher value indicates that higher representations are sent to the edge-servers. For mobi edge-servers (HD and norm respectively), the average representation index is 2.31 (3.53 and 2.69 respectively). The result shows that mobi edge-servers host representations with lower index while HD edge-servers representations with higher index. This also conforms to the trends of Fig. 5 in the static configuration.



(a) For different network provisioning

(b) For different time with 15 reflectors

Fig. 11. Average user satisfaction: GEN vs. SPC

2) Impact of Type-Specified User Assignment: SCADOOSH includes user assignment, content placement and tree delivery. We would like first to evaluate the importance of the former algorithm, user assignment. Our proposal is to assign users to edge-servers based not only on the channels they watch, but also on the type of device they use. As previously said, more accurate algorithms can be developed. Our goal here is to see whether such very simple implementation can already ensue in a gain of performances.

We refer to the type-specified user assignment mechanism proposed in §V-A as SPC. We compare it to a strategy, which we refer to as GEN, that is more commonly used in CDN: it first tries to assign a user to an edge-server that hosts the same channel within the same area with traffic load lower than the predefined threshold; otherwise, the user is assigned to the least loaded edge-server within the same area with traffic lower than the threshold.

We first show the user distribution by computing the average user downlink capacity on each group of edge-servers. With SPC, the average user downlink is 0.97 Mbps for mobi edge-servers, 23.68 Mbps for HD, and 2.01 Mbps for norm. With GEN, all edge-servers have almost a similar average user downlink capacity (the value is around the average value 13.6 Mbps). This confirms that SPC assigns users to edge-servers based on their capacity.

We plot in Fig. 11(a) the average satisfaction of users for both user assignment algorithms. As expected, SPC performs better, but the overall gain is not formidable: less than 3% of satisfaction in average. We show in Fig. 11(b) that the average user satisfaction changes during the simulation time (with 15 reflectors). We first observe that SPC always get a better average user satisfaction than GEN. We also compare user population changes in Fig. 8. Here, users are less satisfied when the system serves more users because more channels are requested at each edge-server so the system becomes more underprovisioned.

At last, we show the gains of SPC more obviously in Fig. 12 and Fig. 13 for two CDN configurations (15 and 20 reflectors respecively). We observe here that SPC is able to deliver the best representation to a larger subset of the population: 70% of the population instead of 61% for 15 reflectors; and 86% of the population instead of 76% for 20 reflectors. With regard to the low complexity of the SPC algorithm, such gain can justify its implementation.

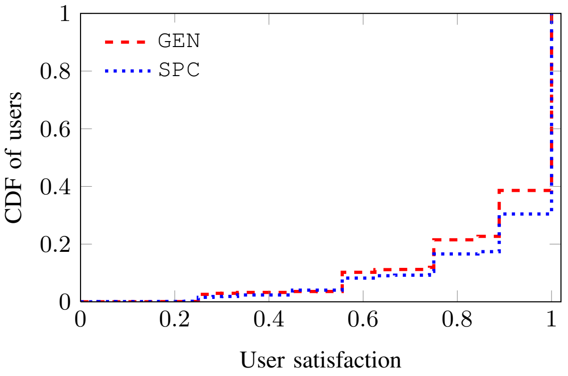


Fig. 12. CDF of user satisfaction with 15 reflectors

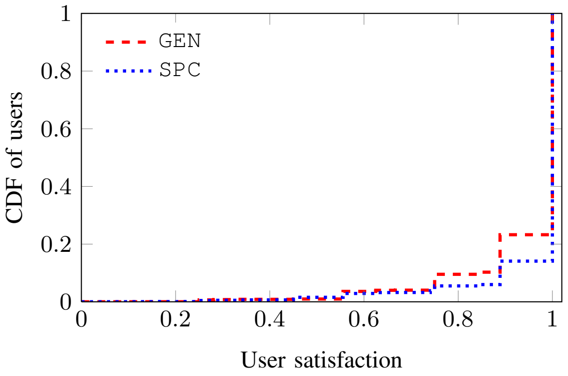


Fig. 13. CDF of user satisfaction with 20 reflectors

VII. Related work

A surprisingly low amount of work addresses live video streaming in CDN networks. The main set of publications (e.g. [4,5,7,8]) aim to reduce the transmission cost under resource constraints in a context of inter-domain Internet architecture. However, none of them targets for rate-adaptive streaming. As we stated in Sec. I, the multi-representations characteristic of rate-adaptive streaming imposes capacity problem on the CDN infrastructure, while at the same time, it brings opportunities for novel delivery strategies.

As previously said, some related work addressed the problem of maximizing the bit-rate of data stream, subjects to the upload capacities of nodes in peer-to-peer networks [10–13]. The objective is to maximize the throughput of the network by maximizing the bit-rate of the stream. The formulation of the problem under various network conditions and application scenarios is available in [10]. A network coding based algorithm that obtain near-optimal performances in terms of bandwidth utilization is proposed in [13]. Unfortunately, these solutions are unrealizable in a CDN delivering rate-adaptive live videos. Furthermore, the models used in these works are based on elastic video bit-rate. However, in the context of live rate-adaptive video streaming, the bit-rates of representations are well pre-defined, and streams have to be either delivered in its entirety, or not delivered at all. Hence, rather than maximizing the bit-rate of streams, the number of delivered streams should be maximized. We compared our user-centric discretized streaming model to the one that maximizes the deliverable bit-rate, please refer to Sec. IV for more details. A dense literature deals with the delivery of *layered* streams. The context of adaptive streaming is however different since users request only one representation. Moreover representations are independent and have different bit-rates. Recent works have shown that coupling layered video with adaptive streaming [34] can generate important bandwidth savings but such solution is still in its infancy today. Finally, a discretized streaming model for live rate-adaptive video streaming has been proposed in [35]. The goal is to maximize the utility of delivered streams. Our objective is more concrete: we aim to maximize user satisfaction on their perceived video quality in a dynamic environment.

Some works solved the rate control problem for multicast sessions that maximize the bit-rate (or the utility of bit-rate) of multiple multicast sessions [36,37]. Again, the problem is formulated based on elastic bit-rates and the set of multicast trees are given. Rather, in our work, we aim to construct a set of delivery trees that achieves our optimization goal. Numerous work have studied multi-tree packing for peer-to-peer application-layer multicast protocols (see [38] for a survey). The goal is to span all nodes under application-related optimization objective (*e.g.*, to minimize tree height, or to reduce controlling overhead). Bounded tree packing delivery for live video have been studied in the context of peer-assisted systems [39]. The problem is to minimize the amount of additional resources that are needed to guarantee that all peers in a peer-to-peer system can achieve a given download bit-rate. This problem is different in CDNs, which are self-sustained networks; missing resources cannot be compensated, but rather need to be used in the best possible way.

Our multi-tree delivery is also close to various theoretical works. In particular, the minimum Bounded Degree Spanning Tree (BDST) problem aims to determine a minimum-cost spanning tree while no node should have more than m children (see [40]). This problem is NP-complete for any $m \geq 2$. Related variations of this problem feature non-uniform degree bounds [41] in a model that is close to ours. Though, our problem formulation differs since, first, we consider an unweighted graph, and second, these works aim at spanning all nodes in the network while optimizing an objective function, while we aim at maximizing the number of spanned nodes under a node degree constraint. The only related work in this aspect is [42], which studies the minimum spanning tree with at least k nodes in a weighted graph, but this work does not target the maximal k. Furthermore, these works do not deal with packing several trees and the resource allocation problem that such packing introduces.

VIII. Conclusion

This paper is a first step towards user-centric delivery of live rate-adaptive videos in CDN. It provides both a theoretical background and a practical implementation. From a theoretical perspective, our main contribution is the formulation of an optimization problem using the user-centric discretized streaming model. In this problem, we maximize the satisfaction of users subject to the underprovisioning of CDN infrastructure. We formulate an Integer Linear Program

(ILP), which jointly decides the choice of representations that should be sent to the edge-servers, the building of a delivery overlay, and the assignment of end-users to edge-servers. We show that previous works related to streaming capacity do not satisfactorily address the problem met by CDN with rate-adaptive streaming.

Our second contribution is the design of a practical system, namely SCADOOSH. It features a coordinator that periodically executes three algorithms, which are directly inspired from our theoretical analysis: a user assignment algorithm, a content placement algorithm based on the computation of stream utility, and a delivery tree construction algorithm. We evaluate the performances of SCADOOSH through a dynamic large-scale trace-based simulation. We show that SCADOOSH can maintain a good Quality of Experience even when the CDN infrastructure is severely underprovisioned. In addition, SCADOOSH improves the number of users that are served with their best representations.

We envision multiple future works to this paper. We would like to enhance the optimization problem and extend our study of the delivery algorithms in two directions. We are especially interested in delivery algorithms for packs of trees, and solutions for the resource allocation problem that such packing introduces. The design of polynomial-time algorithms with proven performance guarantees is also a major objective since it could have some practical consequences. We are also interested in determining families of network infrastructure on top of which fast, optimal delivery algorithms can be built. Such works can probably be inspired by the vast literature related to minimum Bounded Degree Spanning Tree (BDST) problems.

In a more practical perspective, SCADOOSH is a first step, which deserves further studies. We will investigate each of the three algorithms that are presented in the current paper to enhance the necessary trade-off between simplicity of implementation and practical benefits. One of the most interesting challenges is to revisit forecasting algorithm so that the multi-representations feature of rate-adaptive streaming is actually taken into account. Another challenge is to determine some network topologies on top of which the QoE can be even better for a given underprovisioning. Finally, we are discussing on implementing SCADOOSH on a real-world CDN infrastructure.

References

[1] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, “Unreeling netflix: Understanding and improving multi-cdn movie delivery,” in *IEEE INFOCOM*, 2012, pp. 1620–1628.

[2] M. Ingram, “You think the internet is big now? akamai needs to grow 100-fold,” Om Malik, Jun 2012, http://is.gd/3vTZPC.

[3] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, “Inter-datacenter bulk transfers with netstitcher,” in *QCM SIGCOMM*, 2011.

[4] M. Adler, R. K. Sitaraman, and H. Venkataramani, “Algorithms for optimizing the bandwidth cost of content delivery,” *Computer Networks*, vol. 55, no. 18, pp. 4007–4020, 2011.

[5] K. Andreev, B. Maggs, A. Meyerson, J. Saks, and R. Sitaraman, “Algorithms for constructing overlay networks for live streaming,” *CoRR*, 2011.

[6] F. Zhou, S. Ahmad, E. Buyukkaya, G. Simon, and R. Hamzaoui, “Minimizing Server Throughput for Low-Delay Live Streaming in Content Delivery Networks,” in *ACM NOSSDAV*, 2012.

[7] K. Andreev, B. M. Maggs, A. Meyerson, and R. K. Sitaraman, “Designing overlay multicast networks for streaming,” in *ACM SPAA*, 2003.

[8] E. Nygren, R. K. Sitaraman, and J. Sun, “The Akamai network: a platform for high-performance internet applications,” *Op. Sys. Rev.*, vol. 44, no. 3, pp. 2–19, 2010.

[9] “Netflix Open Connect Peering Guidelines,” 2012, https://signup.netflix.com/openconnect/guidelines.

[10] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. A. Chou, “Peer-to-peer streaming capacity,” *IEEE Trans. Inf. Theory*, 2011.

[11] J. Kim and R. Srikant, “Achieving the maximum p2p streaming rate using a small number of trees,” in *IEEE ICCCN*, 2011.

[12] C. Zhao, X. Lin, and C. Wu, “The streaming capacity of sparsely-connected P2P systems with distributed control,” in *IEEE INFOCOM*, 2011.

[13] D. Niu and B. Li, “Asymptotic optimality of randomized peer-to-peer broadcast with network coding,” in *IEEE INFOCOM*, 2011.

[14] [Online]. Available: http://en.justin.tv/

[15] A. Reis, J. Chakareski, A. Kassler, and S. Sargento, “Distortion optimized multi-service scheduling for next-generation wireless mesh networks,” in *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, 2010, pp. 1–6.

[16] S. Thakolsri, W. Kellerer, and E. Steinbach, “Qoe-based rate adaptation scheme selection for resource-constrained wireless video transmission,” in *Proc. international conference on Multimedia*, 2010.

[17] A. Khan, L. Sun, E. Jammeh, and E. Ifeachor, “Quality of experience-driven adaptation scheme for video applications over wireless networks,” *Communications, IET*, vol. 4, no. 11, pp. 1337–1347, 2010.

[18] Z. Ma, H. Hu, M. Xu, and Y. Wang, “Rate model for compressed video,” *CoRR*, vol. abs/1206.2625, 2012.

[19] K. Yamagishi and T. Hayashi, “Parametric packet-layer model for monitoring video quality of iptv services,” in *IEEE ICC*, 2008, pp. 110–114.

[20] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, “Qoe-driven cache management for http adaptive bit rate streaming over wireless networks,” *IEEE Trans. Multimedia*, 2013.

[21] Apple, “Using http live streaming,” http://goo.gl/fJIwC.

[22] K. Mitra, A. B. Zaslavsky, and C. Åhlund, “A probabilistic context-aware approach for quality of experience measurement in pervasive systems,” in *Proc. ACM SAC*, 2011.

[23] A. Rao, Y.-S. Lim, C. Barakat, A. Legout, D. Towsley, and W. Dabbous, “Network Characteristics of Video Streaming Traffic,” in *Proc. ACM CoNEXT*, 2011.

[24] R. Andrews, “Nbc: Nearly half of olympics streams are from mobile, table,” http://is.gd/ZlJnaJ.

[25] “GeoDNS BIND patch,” http://www.caraytech.com/geodns/.

[26] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, “Dissecting Video Server Selection Strategies in the YouTube CDN,” in *IEEE ICDCS*, 2011.

[27] L. Plissonneau and E. Biersack, “A Longitudinal View of HTTP Video Streaming Performance,” in *MMSys*, 2012.

[28] V. K. Adhikari, S. Jain, and Z.-L. Zhang, “Where Do You “Tube”? Uncovering YouTube Server Selection Strategy,” in *IEEE ICCCN*, 2011.

[29] “The value of smart pipes to mobile network operators,” Tellabs STL Reports, Dec. 2011, http://info.tellabs.com/smartpipes.

[30] F. Wang, J. Liu, and M. Chen, “Calms: Cloud-assisted live media streaming for globalized demands with time/region diversities,” in *IEEE INFOCOM*, 2012, pp. 199–207.

[31] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[32] C. Wu, B. Li, and S. Zhao, “On dynamic server provisioning in multichannel p2p live streaming,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1317–1330, 2011.

[33] J.-H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, 2004.

[34] M. Grafl, C. Timmerer, H. Hellwagner, W. Cherif, and A. Ksentini, “Evaluation of hybrid scalable video coding for HTTP-based adaptive media streaming with high-definition content,” in *Proc. of IEEE WoWMoM Workshop*, 2013.

[35] J. Liu and G. Simon, “Fast Near-Optimal Algorithm for Delivering Multiple Live Video Streams in CDN,” in *IEEE ICCCN 2013*, 2013.

[36] M. R. Rahimi, A. Bais, and N. Sarshar, “On fair and optimal multi-source ip-multicast,” *Computer Networks*, vol. 56, no. 4, pp. 1503–1524, 2012.

[37] K. Kar, S. Sarkar, and L. Tassiulas, “Optimization based rate control for multirate multicast sessions,” in *IEEE INFOCOM*, 2001, pp. 123–132.

[38] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, “A survey of application-layer multicast protocols,” *IEEE Commun. Surveys & Tut.*, vol. 9, no. 3, pp. 58–74, 2007.

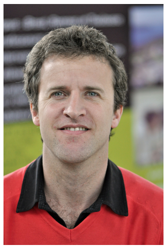
[39] R. Sweha, V. Ishakian, and A. Bestavros, “AngelCast: Cloud-based Peer-Assisted Live Streaming Using Optimized Multi-Tree Construction,” in *ACM MMSys*, 2012.

[40] M. X. Goemans, “Minimum bounded degree spanning trees,” in *IEEE FOCS*, 2006.

[41] J. Könemann and R. Ravi, “Primal-dual meets local search: approximating MSTs with nonuniform degree bounds,” *SIAM J. Comput.*, vol. 34, no. 3, pp. 763–773, 2005.

[42] C. Blum and M. J. Blesa, “New metaheuristic approaches for the edge-weighted -cardinality tree problem,” *Computers & OR*, vol. 32, pp. 1355–1377, 2005.

Jiayi Liu is a PhD student at Telecom-Bretagne, Rennes, France. She is working, under the supervision of Prof. Gwendal Simon. She obtained a Master of Science in Computer Science from Rennes1 University in 2009. She obtained a Bachelors of Science in Electronic Engineering from Xidian University in Xi’an China, in 2007. Her research interests include peer-to-peer streaming systems, live video streaming in Content Distribution Network, rate adaptive streaming technique, and bandwidth efficient streaming solutions.



Gwendal Simon received his Master Degree in 2000 and his PhD degree in December 2004 both in Computer Sciences from University of Rennes (France). During his PhD, he also worked at Orange Labs. From 2004 to 2006 he was a researcher at Orange Labs. Since 2006, he has been Associate Professor at Telecom Bretagne, a graduate engineering school within the Institut Mines-Telecom. He was a visiting researcher at University of Waterloo from September 2011 to September 2012. His research interests include large-scale distributed networks, peer-to-peer and ad-hoc networks, optimization problems and video delivery systems.

Catherine Rosenberg is a Professor in Electrical and Computer Engineering at the University of Waterloo. Since June 2010, she holds the Tier 1 Canada Research Chair in the Future Internet. From 1999 to 2004, Prof. Rosenberg was a Professor in the School of Electrical and Computer Engineering at Purdue University. Prof. Rosenberg was elected an IEEE Fellow in 2011 and a Fellow of the Canadian Academy of Engineering in 2013. Her research interests are mainly in two areas: the Internet and Energy Systems. Prof. Rosenberg obtained her “Diplome d’Ingénieur” from the Ecole Nationale Supérieure des Télécommunications de Bretagne, her M.S. in Computer Sciences from the University of California at Los Angeles and her “Doctorat en Sciences” from Université Paris XI.

Geraldine Texier is an associate professor at the Network, Security and Multimedia department at Telecom Bretagne, Rennes, France. She is a member of ReOP research team from IRISA. She received her PhD degree on the management of the cooperation and the awareness in computer supported collaborative work in 2000 from the University of Rennes 1 (France). Since 2001, her research activities are focused on routing, in particular on multi-constrained and QoS routing, on inter-domain routing, and on network measurement.