# Representation selection problem: optimizing video delivery through caching

Andrea Araldo*†, Fabio Martignon*§ and Dario Rossi†
* LRI, Université Paris-Sud, {first.last}@lri.fr
† Telecom ParisTech, {first.last}@enst.fr § IUF, Institut Universitaire de France

***Abstract*—To cope with Internet video explosion, recent work proposes to deploy caches to absorb part of the traffic related to popular videos. Nonetheless, caching literature has mainly focused on network-centric metrics, while the quality of users' video streaming experience should be the key performance index to optimize. Additionally, the general assumption is that each user request can be satisfied by a single object, which does not hold when multiple representations at different quality levels are available for the same video.**

**Our contribution in this paper is to extend the classic *object placement* problem (which object to cache and where) by further considering the *representation selection* problem (i.e., which quality representation to cache), employing two methodologies to tackle this challenge. First, we employ a Mixed Integer Linear Programming (MILP) formulation to obtain the centralized optimal solution, as well as bounds to natural policies that are readily obtained as additional constraints of the MILP. Second, from the structure of the optimal solution, we learn guidelines that assist the design of distributed caching strategies: namely, we devise a simple yet effective distributed strategy that incrementally improves the quality of cached objects. Via simulation over large scale scenarios comprising up to hundred nodes and hundred million objects, we show our proposal to be effective in balancing user perceived utility vs bandwidth usage.**

***Keywords*—Content Distribution; Optimization; Quality of Experience (QoE); Caching**

## I. Introduction

The large majority of the Internet traffic currently consists of video delivery. The related traffic is expected to explode due to increasing demand on the one hand, but, more importantly, in reason of the increasing quality expectations of users. Indeed, at the Consumer Electronics Show in Las Vegas, "Beyond 4K Ultra HD" technologies were shown that increase pixel density by 167% [1] over the previous year – a much faster growth rate with respect to worldwide user population.

Caching video content, with either current Content Distribution Network (CDN) technologies and their interconnection [14] or more futuristic and pervasive Information Centric Network (ICN) architectures, may help containing this traffic deluge. However, the caching literature has, with few exceptions [3], considered network-centric metrics like hit-ratio, hit-distance, server offload, etc., overlooking more important aspects related to the quality of user experience.

More importantly, except for some recent effort, video streaming and caching have been mostly studied as orthogonal problems, often in different research communities. Rephrasing the title of [12], caching and video are still not friends: classic video streaming mechanisms assume that a client downloads a video from a single source, which is not true in presence of caching, misleading control loops. Moreover, caching techniques are designed with generic content in mind, whereas we show in this paper that video traffic has peculiarities that demand for caching mechanisms specifically tailored for it. The most important of these peculiarities is a different request-to-object mapping assumption: previous studies assume that a user request can be mapped to a single object, while a request for a video can be served by providing one of the different representations of the same video, corresponding to different quality levels, and ultimately different levels of user satisfaction.

As a first consequence, it is no longer sufficient to choose which *object* to cache, but also which of its available *representations*. Therefore, we add a new dimension to caching techniques: in addition to the classic *object placement* problem, i.e., which object to cache and where, we also consider the *representation selection* problem, i.e. which quality representation to cache. As a second consequence, the bandwidth required to satisfy a certain request is no more univocally determined by the object identifier, but depends on the quality at which we decide to serve that request. ISPs can leverage the possibility of serving the same request by using different bandwidth amounts to efficiently exploit their links and adapt to the dynamics of traffic, maximizing user satisfaction at the same time.

We consider a scenario in which Autonomous Systems (AS) peer together forming a coalition to collaboratively share their cache resources. We do not investigate the coalition formation problem, and rather focus on providing a strategy for the AS coalition to maximize the quality perceived by their users. Our key contributions can be summarized as follows:

- We propose a novel representation-aware Mixed Integer Linear Programming (MILP) model, which determines the object placement, quality representation selection and routing, taking into account video quality in order to maximize users' experience, in a capacity and cache size-constrained network scenario.
- The knowledge gained by studying the structure of the optimal solution inspires the design of a distributed caching strategy, which we implement in an event-driven simulator to scale up the analysis to network sizes of hundred nodes and catalog of hundred million objects.

Our key finding is that, despite the cache deployment considerably helps in improving user quality of experience, utility maximization can be achieved by (i) minimizing the number of representations stored per object (to increase the cache efficiency), and (ii) selecting the most useful representation for

each object (which is at the heart of the representation selection problem). We thus devise a simple yet effective distributed strategy that: (i) maintains a single representation per object, and (ii) incrementally improves the quality of cached objects at each new request, so that the average quality in steady state is inversely related to the object popularity.

This paper is structured as follows. Sec. II casts this work in the context of related effort. Sec. III introduces the methodologies used in our work, extensively describing (i) the representation-aware MILP model and its variants, as well as (ii) the online distributed cache algorithm. Sec. IV illustrates our numerical and simulation results in both (i) toy case scenarios, to understand properties of the optimal solution as well as (ii) large scale scenarios, to confirm our reasoning to hold in more general cases. Sec. V concludes the paper with a summary of our findings and our future work agenda.

## II. Related work

Video streaming over the Internet has become a mainstream research topic in recent years: as such, several works focused on the problem of ensuring an efficient *video streaming* in communication networks. Similarly, *caching* is a very effective technique that permits to serve contents in both bandwidth and time-efficient manners, which has attracted a surge of attention in recent years through popularization of Content Distribution and Information Centric networks. However, as already discussed, there is still lack of a unified viewpoint to alleviate the huge increase in required bandwidth and guarantee satisfactory Quality of Experience (QoE) for users.

To confirm this, classic caching directly applied to video streaming is not only inefficient but can even be harmful [12]. Another example of classic caching vs. classic video streaming impairment is given in [16], which, by means of trace-driven simulations, finds that an ICN cache deployment would not lead to relevant QoE improvement in video delivery. Yet we argue that such results understate the benefits achievable via caching, since they are obtained by applying representation-blind policies, which consider homogeneous objects, all encoded at a single quality. In this work, we instead leverage the possibility to serve different quality representations to maximize user satisfaction, respecting capacity constraints.

Conversely, QoE maximization has been tackled in the classic video literature [6], [12], [13] by proposing control mechanisms that intelligently share bandwidth among different users. Control algorithms in scenarios with multiple sources (like caches and repositories) are proposed by [12], [13]. In particular, the former shows that quality fluctuations can be observed because of caching, which hampers QoE. Both works evaluate control algorithms under a given content allocation, whereas we look for the allocation guaranteeing the best QoE. Authors of [15] consider caching of videos in a heterogeneous network, assuming that users can specify the minimum video quality they are willing to accept and the network provider goal is to minimize delay and cost while providing at least that quality. Our viewpoint is different, since we directly measure user satisfaction in terms of quality provided, rather than delay, and our goal is not just to satisfy a minimum requirement but to send videos at the maximum possible quality. In a similar context, the work in [20] introduces a new layered video encoding, while our enhancement is obtained using the currently most deployed technologies, like MPEG-DASH. Moreover, the context of our model is a multi-AS environment, where the capacity of multi-hop paths limits the rate of transmission (thus, the served quality) whereas in wireless contexts the limitation is due to the channel condition. The closest work to ours is perhaps [10], which employs caching, transcoding and routing functions to minimize the networking cost in a video distribution context. A two-step iterative approach is proposed, where, first, storage and computing resources are allocated optimally, then the routing is configured in the second phase. However, the model does not explicitly account for the utility perceived by users downloading different video representations, which is the focus of our paper.

The fact that a single video can be represented at different qualities has an important impact on users' quality of experience, which [18] and [21] study in a CDN and wireless scenario, respectively. Both works investigate what is the subset of video quality levels to make available in order to maximize QoE, yet both make crude simplifications of the network settings: the former characterizes a delivery system only by the total bandwidth, while the latter only considers one cache and one video. Differently, we assume that the set of quality levels is already established, and look at the problem from a network viewpoint.

## III. Methodology

This section explains our methodologies, casting them to an AS-level system model (Sec.III-A). We formulate a Mixed Integer Linear Programming (MILP) model that maximizes the users' quality of experience, taking into accurate account the different object representations available, as well as capacity and cache constraints (Sec. III-B), discussing its limits and possible extensions (Sec. III-C). We constrain our model to give solutions with simpler structures guaranteeing, at the same time, performance close to the optimum (Sec. III-D). The solution of the MILP, that we report in a later section, then guides the design of an effective distributed caching policy that can be easily implemented in practice (Sec. III-E). Tab. I summarizes the notation used throughout this paper.

### A. System model

We illustrate the system model considered in our work, with the help of an example scenario depicted in Fig. 1. We consider a set $V = \{1, \dots, V\}$ of Autonomous Systems (ASes), whose interconnection is represented by a graph, composed of nodes and capacitated arcs. Nodes in the graph (ASes) can act as *content producers* (when they are directly connected to some repositories), *transit* ASes that merely participate in the content caching and diffusion, or *consumer* ASes that additionally generate video requests. Repositories and caches distributed in the ASes store objects (in particular, multimedia content and videos), of which different representations (quality levels) exist, belonging to a discrete set $Q$. Each of these quality levels is associated to a rate $r^q$ necessary to support and transmit the object at the given quality $q$, as well as to a storage space $s^q$ that is necessary to cache it. AS users issue requests for videos without specifying the quality representation, given that the model will find the optimal one.

Each AS has *upstream links* through which data is retrieved from other nodes and *downstream links* through which data is

Table I. SUMMARY OF THE NOTATION USED IN THIS PAPER.

| Parameters of the Models | |
|---|---|
| $A$ | Set of arcs |
| $V$ | Set of Nodes (Autonomous Systems, ASs) |
| $O$ | Set of objects |
| $Q$ | Set of qualities |
| $FS(i)$ | Set of forward arcs $(i,j) \in A$ for node $i \in V$ |
| $BS(i)$ | Set of backward arcs $(j,i) \in A$ for node $i \in V$ |
| $b_e$ | Capacity of the arc $e \in A$ |
| $n_v^o$ | Number of requests for object $o$, in AS $v \in V$ |
| $r^q$ | Rate required to retrieve an object at quality $q \in Q$ |
| $s^q$ | Storage space required to cache an object at quality $q \in Q$ |
| $U^q$ | Utility gained to serve one request for an object at quality $q$ |
| $p_v^o$ | 0-1 Producers reachability matrix<br>$p_v^o = 1$ if AS $v$ has a producer for object $o \in O$<br>(it can serve whatever quality of object $o$) |
| $S_v$ | Max caching storage that can be installed at AS $v$ |
| $S_{TOT}$ | Max caching storage that can be installed in the network |
| $bw_v$ | Max egress capacity for AS $v \in V$,<br>$bw_v = \max\left(\sum_{e \in FS(v)} b_e; \sum_{o \in O} n_v^o \cdot \max_{q \in Q} r^q\right)$ |

| Decision Variables of the Models | |
|---|---|
| $n_v^{o,q}$ | Number of requests for object $o$ at quality $q$ satisfied at AS $v$ |
| $x_{v_s}^{o,q}$ | 0-1 Caching variable, if the source AS $v_s \in V$ caches $o$ at quality $q$ |
| $y_e^{o,q,v_d}$ | Flow on arc $e \in A$ for object $o \in O$, at quality $q$ sent to the destination AS $v_d \in V$ |
| $d^{o,q,v_d}$ | Rate requested at AS $v_d \in V$, for object $o$ at quality $q$ |
| $z_{v_s}^{o,q,v_d}$ | Rate provided by the source AS $v_s \in V$, for object $o$, at quality $q$ for the destination $v_d \in V$, when $v_s$ behaves as a producer ($p_{v_s}^{o,q} = 1$) |
| $w_{v_s}^{o,q,v_d}$ | Rate provided by the source AS $v_s \in V$, for object $o$, at quality $q$ for the destination $v_d \in V$, when $v_s$ behaves as a cache ($x_{v_s}^{o,q} = 1$) |

sent to users. ASes are endowed with caching capabilities, and can store objects as well as route object requests/data towards neighbor routers, the repository or clients. To be as general as possible, we do not specify the details of the technology that provides caching capabilities (ICN, CDN, Web proxy, etc.). Each object can be served at different qualities, which may depend on the network characteristics (link capacities, bottlenecks) and the clients position, and produce a utility that is experienced by users. The aim of our work is to determine (i) optimal allocations of objects to AS caches, (ii) optimal quality level(s) to store for each cached object and to map to each request, as well as (iii) optimal routing strategies, that collectively contribute in maximizing the overall utility perceived by network users.

### B. Representation-Aware MILP

The Representation-Aware model that maximizes users' utility can be formalized as follows:

$$\max \sum_{o \in O} \sum_{q \in Q} \sum_{v \in V} n_v^{o,q} U^q \tag{1}$$

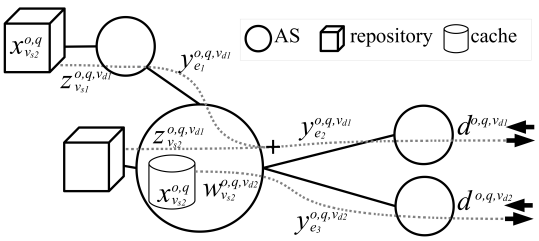



Figure 1. Example scenario indicating the main variables employed in the MILP model.

subject to:

$$\sum_{q \in Q} n_v^{o,q} = n_v^o \qquad \forall o \in O, v \in V \tag{2}$$

$$d^{o,q,v_d} = n_{v_d}^{o,q} \cdot r^q \qquad \forall o \in O, q \in Q, v_d \in V \tag{3}$$

$$d^{o,q,v_d} = z_{v_d}^{o,q,v_d} + w_{v_d}^{o,q,v_d} + \sum_{e \in BS(v_d)} y_e^{o,q,v_d} - \sum_{e \in FS(v_d)} y_e^{o,q,v_d} \qquad \forall o \in O, q \in Q, v_d \in V \tag{4}$$

$$z_{v_s}^{o,q,v_d} + w_{v_s}^{o,q,v_d} + \sum_{e \in BS(v_s)} y_e^{o,q,v_d} = \sum_{e \in FS(v_s)} y_e^{o,q,v_d} \qquad \forall o \in O, q \in Q, v_s \in V, v_d \in V, v_s \neq v_d \tag{5}$$

$$\sum_{o \in O} \sum_{q \in Q} \sum_{v_d \in V} y_e^{o,q,v_d} \leq b_e \qquad \forall e \in A \tag{6}$$

$$\sum_{v_d \in V} z_{v_s}^{o,q,v_d} \leq p_{v_s}^{o,q} \cdot bw_{v_s} \qquad \forall o \in O, q \in Q, v_s \in V \tag{7}$$

$$\sum_{v_d \in V} w_{v_s}^{o,q,v_d} \leq x_{v_s}^{o,q} \cdot bw_{v_s} \qquad \forall o \in O, q \in Q, v_s \in V \tag{8}$$

$$\sum_{o \in O} \sum_{q \in Q} x_{v_s}^{o,q} \cdot s^q \leq S_{v_s} \qquad \forall v_s \in V \tag{9}$$

$$\sum_{o \in O} \sum_{q \in Q} \sum_{v_s \in V} x_{v_s}^{o,q} \cdot s^q \leq S_{TOT} \tag{10}$$

$$x_v^{o,q} \in \{0,1\} \qquad \forall o \in O, q \in Q, v \in V \tag{11}$$

$$n_v^{o,q} \in \mathbb{Z}^+ \qquad \forall o \in O, q \in Q, v \in V \tag{12}$$

$$y_e^{o,q,v_d} \in \mathbb{R}^+ \qquad \forall o \in O, q \in Q, v_d \in V, e \in A \tag{13}$$

$$d^{o,q,v_d} \in \mathbb{R}^+ \qquad \forall o \in O, q \in Q, v_d \in V \tag{14}$$

$$z_{v_s}^{o,q,v_d}, w_{v_s}^{o,q,v_d} \in \mathbb{R}^+ \qquad \forall o \in O, q \in Q, v_d \in V, v_s \in V. \tag{15}$$

In particular, objective function (1) represents the overall utility experienced by network users, which is maximized by our model. The set of constraints (2) makes sure that all the requests are served at one (or more) quality level(s). In the problem instances we add a "special" quality level $q = 0$, which represents unserved traffic demands: when serving quality $q = 0$, no bandwidth is required ($r^q = 0$); moreover, no utility is generated, $U^0 = 0$. Constraints (3)

set the value of the rate requested at AS $a$, for object $o$, at quality $q$. Such demand is satisfied in (4). In particular, it can be satisfied because: (i) the AS is a producer for that object (i.e.: $z_{v_d}^{o,q,v_d} = d^{o,q,v_d}$), (ii) the AS caches the object (i.e.: $w_{v_d}^{o,q,v_d} = d^{o,q,v_d}$), or (iii) the AS retrieves the object (i.e.: the sum of flows on incoming links).

Flow balance constraints are imposed in (5) and we bound the arc capacity in (6). Similarly, in (7) and (8), we limit the maximum emitted flows the AS sends when it behaves as a producer and a cache, respectively. The overall caching storage that can be deployed by an AS is bounded in (9), and we extend the same limit to the entire topology in (10). Finally, integrality and non-negativity constraints are imposed in (11)-(15).

### C. Discussion

The problem that our model aims to solve can be conceptually formulated as follows: at a generic time instant we have facilities, i.e. link capacities and caches and a set of concurrent user requests for video chunks. Our goal is to find the facility allocation that maximizes users utility. In other words, we adopt a snapshot approach, as usually done in optimization works, which is based on this instantaneous picture of the system. Although this might be considered too simplistic, almost all the vast and notable literature, e.g. [5], [8], [10], [15], [18], [20], [21], which applies optimization models to network analysis is based on it, even when not explicitly stated, and results have been widely accepted by the community. For these reasons, the plausibility of the snapshot approach is unlikely to be questionable and, however, we build on it only to show meaningful insights on the novel representation selection problem, rather than to provide absolute measures. On the other hand, we analyze realistic scenarios, where requests can arrive at any moment and the system evolves from time to time, in Sec. IV-E by means of simulation.

While related work usually aims to minimize delay in order to improve user perception, we focus instead on maximizing the provided quality for two reasons: i) we want our contribution to be complementary to this related work, ii) the packet delay can be absorbed by playout buffers and be invisible to the user. The only exception to this is when this delay is excessively high or variable, causing high startup times or rebuffering episodes. This happens in case of congestion. For these reasons, rather than looking at the delay, we focus on caching content at the right quality, such that it can be transmitted using the available bandwidth on the path, thus avoiding congestion.

Another aspect worth underlining is that in today's video delivery, plugins in the user Web browser select the quality representation to request, while we assume that ISPs choose the best possible quality to serve its users. This is not unrealistic since, in either case, users do not make any explicit choice most of the time [9], so that the selection mechanism, be it done in the Web browser of their personal device, or at the proxy in the ISP premises, is completely transparent to them.

Additionally, we remark an increasing tendency toward adaptive streaming quality, which is however not massively deployed or still in early stage even for big Internet players [2]. These details are related to the implementation of congestion control algorithms that are however outside the scope of this work. On the contrary, our study focuses on caching and aims at finding the performance bounds from a more abstract viewpoint. The findings we provide here should be considered what an optimal caching strategy can theoretically achieve, supposing a perfect congestion control mechanism at the bottom. For this reason, we can adopt the snapshot approach, as in other notable works on video delivery [18].

It is worth noticing that our model can be easily extended to have a fine-grained representation, considering heterogeneity of video type and user device. As for the former, it is known that videos with different subjects (sport, movies, TV shows), even if encoded at the same bit-rate and resolution, are perceived in a different way [18]. As for the latter, a user watching a video on a smartphone may be perfectly satisfied with a resolution and a bit-rate lower than the one demanded by a user using a ultra-HDTV 4K screen. However, this level of detail is beyond the scope of this paper and such directions can be incorporated at a later step in the model.

Table II. MODEL VARIANTS IMPLEMENTING NATURAL AS POLICIES: ADDITIONAL CONSTRAINTS FOR THE MILP MODEL (1)-(15)

| Caching Policy | Additional constraint in MILP model |
|---|---|
| *NoCache* | $x_v^{o,q} = 0$ |
| *CacheLQ* | $x_v^{o,q} = 0, \forall q \neq LQ$ |
| *CacheHQ* | $x_v^{o,q} = 0, \forall q \neq HQ$ |
| *AllQ* | $x_v^{o,q_h} = x_v^{o,q_k}, \forall q_h, q_k \in Q$ |
| *Partitioned* | $\sum_{o \in O} x_{v_s}^{o,q} \cdot s^q \leq S_{v_s} \cdot \frac{s^q}{\sum_{q' \in Q} s^{q'}}$ |

### D. Modeling AS policies

Jointly deciding the optimal representations that each node should cache and serve to users is a hard task to be performed by a distributed online strategy, in which each node makes local decisions without having knowledge of the status of the rest of the network and the overall set of requests. Nonetheless, our final aim is to give a feasible solution that can be deployed in a real network, providing a good performance at least close to the optimal one.

We thus constrain our model to give solutions with a simpler structure and we verify how far they are from the optimum. The constrained variants of the model, detailed hereafter, are easier to approximate in distributed, online algorithms and, as our numerical results will show, some of them exhibit indeed very good performance, close to optimality in several situations. Thanks to the flexibility of our MILP model, modeling AS policies is as simple as adding a single constraint for each strategy.

Such constraints, specified in Tab. II, include: a *No Cache* strategy, which never caches videos; *CacheLQ* and *CacheHQ*, which exclusively cache the lowest (highest) quality representation available, indicated with quality level *LQ* (*HQ*), respectively; *AllQ*, which caches all quality representations for any cached object; finally, *Partitioned* stores the same *number* of objects for each quality representation (while their buffer occupancy depends on the quality of the corresponding representation). Note that the constraints only concern caching, and do not force to serve a request with a specified quality representation. For example, a HQ video can still be served, even when *CacheLQ* is employed; in such case, given that HQ videos cannot be cached, they must be retrieved directly by a

repository and cross all links between this latter and the user. In this work, we assume that all ASes in the coalition use the same policy, chosen among the ones described above.

### E. Distributed policy: Bandwidth-utility trade-off

In the MILP model, our only objective is to maximize the utility. To do so, we let the model use the links at their full capacity. In practice, ASes may tend to limit link utilization, in order to avoid the occurrence of congestion, to ensure low end-to-end latency and bound operational costs associated to traffic transmitted toward transit providers. Therefore, a trade-off arises between the utility provided to users and the bandwidth used to guarantee it, that we investigate via simulation.

The bounds of this trade-off are represented by two scenarios: namely *OnlyLQ* (*OnlyHQ*) where objects have a single representation equal to the lowest (highest) quality level. These policies respectively correspond to a crude attempt to minimize the bandwidth (vs maximize the utility) but, as a consequence, incur in low user utility (vs high bandwidth usage).

Between these two extremes, we propose a *Quality Improvement* (*QImpr*) strategy that reactively incrementally improves quality of stored replicas at each new request, and opportunely balances bandwidth and user utility. *QImpr* operates as follows. Each request $req(q)$ carries a value $q$ specifying the minimum required quality, which is always set to $q = 1$ at the ingress of the network (either by the user browser or the ISP proxy) meaning that any quality is accepted (i.e., receiving a LQ representation is preferable to not receiving the video at all). When a new request $req(q)$ arrives at any cache, if a copy at quality $q_{\text{cached}} \geq q$ is found, the request is served with that copy and, at the same time, the AS node issues another request $req(q_{\text{cached}}+1)$ for the same object. Otherwise, $req(q)$ is normally forwarded.

Caches maintain objects in an ordered list. Whenever an object $o$ at quality $q$ arrives, if a better quality $q_{\text{cached}} \geq q$ of that object is already cached, the incoming object is discarded. Otherwise, the new object representation (i) is placed at the head of the list, (ii) any lower quality representation of $o$ is evicted, (iii) if further space is needed to store $o$, this is obtained by evicting cached objects starting from the least recently used one, up toward the head, until a sufficient space to accommodate $o$ at quality $q$ is available. Shortly, expected benefits of this policy are that unpopular objects will only be cached at low quality, whereas popular objects will quickly escalate quality levels. On the downsides, popular objects will be requested at multiple quality levels, generating a slight overhead in the quality improvement process.

Note that in reason of size heterogeneity between representations at different levels, caching a new object causes the eviction of a variable number of least-recently-used objects sufficient to make room for the incoming higher quality representation. This is in contrast with what usually assumed in the ICN-flavored caching literature that assumes all chunks having equal size.

## IV. Results

This section evaluates the impact of caching on the overall video quality perceived by users, showing the validity of the caching strategies proposed so far. To this aim, we provide both numerical solutions of the MILP model via the CPLEX 12.5 solver for the centralized policies and the results of discrete event simulation for the distributed solutions. After describing the scenario in Sec. IV-A, we investigate performance and properties of the proposed strategies in an incremental fashion.

Focusing on a single AS, we first illustrate the structure of the optimal solution in Sec. IV-B. We then thoroughly analyze the bandwidth-storage tradeoff in light of variable representations in Sec. IV-C. We next contrast the range of centralized policies, as well as the distributed *QImpr* policy, in Sec. IV-D. Moving to a multi-AS scenario, we finally confirm MILP results to hold on a 10-node topology, and extend the simulation results to cover topologies up to 100-nodes and catalogs up to $10^8$ objects in Sec. IV-E

### A. Scenario

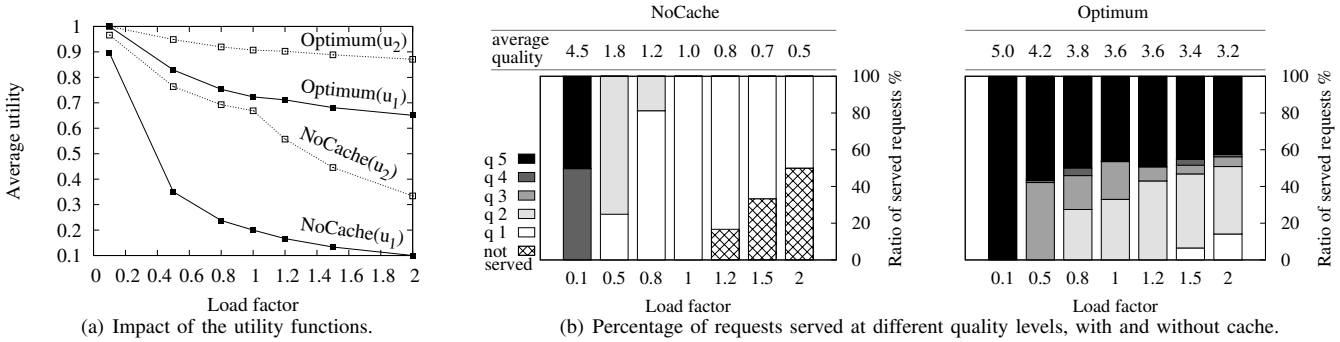
We consider five quality levels [7] in the set $Q$ as reported in Tab. III. Each quality corresponds to a given resolution and bitrate, which both increase for increasing quality levels. We only report the bitrate as this is more pertinent to our optimization goal: video bitrate correlates to both cache storage space, as well as network bandwidth. Resolution, instead, does not come into play directly in the system model, apart from determining a different user perception, that is accounted for in the utility function.

The utility function must be an increasing function of the provided quality, since the higher the quality provided to a user is, the better the utility. Moreover, it must be concave to express the diminishing return in the experience of human vision when providing improved quality [17]. The exact shape of such an utility function is still subject to debate, and there is no unanimously accepted function. However, gathering this function is a hard task that requires intensive experimentation with real users, which is far from the topic of this work. To gather results that are not tied to a specific function, we consider two shapes at the broad end of the spectrum of plausible utility functions, tabulated in Tab. III. Specifically, we define $u_1(q)$ as a model with linear return with respect to the quality: the model likely underestimates contributions of low quality videos, and does not exhibit diminishing returns [17], so that it is biased toward high-quality content. We next define $u_2(q)$ as a power function with a higher concavity: this model does exhibit diminishing returns but sits at the other side of the spectrum as it possibly overestimates contributions of low-quality videos (notice indeed that $u_2(q = 1) > 3u_1(q = 1)$). In the following, we will report the *average system utility* as the average per-request utility, i.e. the total utility as in (1) divided by the total number of requests.

Unless otherwise stated, we consider the single AS scenario depicted in Fig. 3: at a logical level, in the cache-stream we

Table III. Quality levels and corresponding transmission rates, cache occupancy and perceived utility (linear/concave).

| Quality | Rate (Kbps) | Utility $u_1(q)$ | Utility $u_2(q)$ |
|---|---|---|---|
| 1 | 300 | 0.2 | 0.67 |
| 2 | 700 | 0.4 | 0.80 |
| 3 | 1500 | 0.6 | 0.88 |
| 4 | 2500 | 0.8 | 0.95 |
| 5 | 3500 | 1 | 1 |




(a) Impact of the utility functions.

(b) Percentage of requests served at different quality levels, with and without cache.

Figure 2. Single-AS scenario: (a) Benefits of optimal caching and impact of utility function; (b) breakdown of the utility across quality levels for varying load.

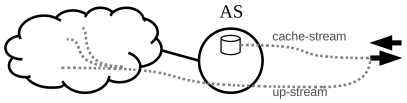



Figure 3. Single-AS scenario: Videos can be downloaded from the cache (cache-stream), while up-stream includes the flows retrieved through other ASes.

include the flows of videos downloaded from the cache, while up-stream includes the flows retrieved through other ASes. The cache represents the aggregate of several cache nodes within the AS, and similarly the up-stream resource represents a single logical link, aggregating all physical links where the request can be satisfied (i.e., all the links except the one where the request is coming from).

Also, unless otherwise stated, the catalog comprises $O = 10^4$ objects whose popularity is distributed as a Zipf with exponent $\alpha = 1$. The cache space at each AS is sufficient to store 1/100 of the catalog objects at the highest quality, *HQ*. Observe that the size of each object depends on its quality, i.e. an object at quality $q$ is $s^{HQ}/s^q$ times smaller than an object at the highest quality, with $s^{HQ}/s^{LQ}$ exceeding one order of magnitude as can be seen in Tab. III.

All links have the same capacity $b$. We express the number of user requests as a *load factor* $L$, i.e. the factor by which we should multiply $b$ in order to transmit all the requested objects at the lowest quality, *LQ*. Otherwise stated, if the load factor is $L = 1$, even if no cache is deployed in the network, we can satisfy all requests at quality LQ, by fully utilizing the network capacity. Notice, however, that due to cache space, it makes sense to consider a normalized load larger than $L > 1$, since part of the endogenous requests can be served from the cache without consuming upstream bandwidth.

### B. Structure of the optimal solution

We first start assessing the dependency of our results on the particular perceptual model in the single cache scenario. We contrast two extremes, namely the optimal solution against the case in which the system is not equipped with caches (so that this latter can sustain a load at most equal to $L = 1$). The average utility is shown in Fig. 2-(a) for both linear $u_1(q)$ and concave models $u_2(q)$: while quantitative results are of course affected by the peculiar function, qualitative results are instead independent of the utility function considered. In particular, the improvement of user experience provided by optimal caching is notable at high load, where caches at the AS absorb a large fraction of the requests, alleviating the impact of the upstream bandwidth limitation. Since the qualitative results between $u_1(q)$ and $u_2(q)$ remain unchanged, and to avoid cluttering the pictures, we only consider the concave profile of $u_2(q)$ in what follows.

A breakdown of the quality levels served to users is reported in Fig. 2-(b), which helps to better understand the structure of the optimal solution – thus, ultimately, where the utility gain comes from. Without any cache, all the delivered videos must cross the upstream link, and the bandwidth is hardly available to transmit them at high quality, unless the input load is particularly low ($L = 1/10$). At high load, the bandwidth is not sufficient to serve all the requests, not even at the lowest quality, and a growing fraction remains unsatisfied. The situation is drastically improved by optimal caching, which stores a significant fraction of videos, and *especially the most popular ones, at high quality*. Since the requests for these videos account for a large part of the overall requests, the upstream link is relieved of a considerable amount of traffic. As a first consequence, we are able to satisfy all the requests coming from users. Moreover, the most popular objects are served at high quality, which as net effect increases the average utility perceived by users.

### C. Contributions of cache and upstream link

To better understand the relative contribution of storage vs bandwidth, we decompose the video flows arriving at users in *cache-stream* and *up-stream*, where the former is the stream of data retrieved from the cache, whereas the latter is the flow coming from the upstream links, as illustrated in Fig. 3. We represent the breakdown of the utility provided by content retrieved from cache vs content retrieved from upstream in Fig. 4, where the sizes of the circles represent the relative contribution of the two utility values. Circles additionally report the quality breakdown of the two contributions.

From Fig. 4 we first observe that, in the scenarios under consideration, the cache is responsible of the most part of the utility (storage circles are bigger than upstream ones), as it stores the most popular objects (thus intercepting a large fraction of traffic) at a furthermore high quality.

Second, an interesting specialization arises between the cache-stream and up-stream: the highest quality levels (darker

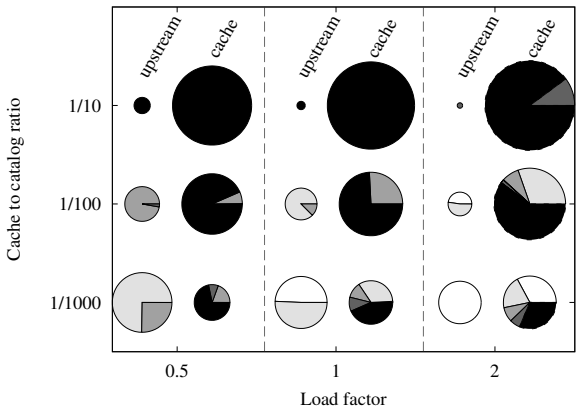



Figure 4. Single-AS scenario: Contributions of cache and upstream links. Circle size reflects relative cache vs upstream contribution, and the breakdown reports the qualities of both contributions.

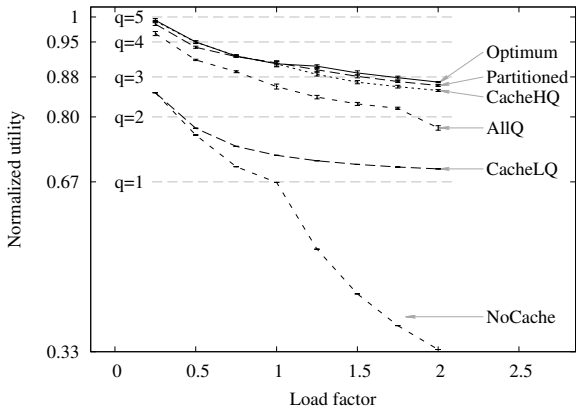



Figure 5. Single-AS scenario: comparison of MILP variants.

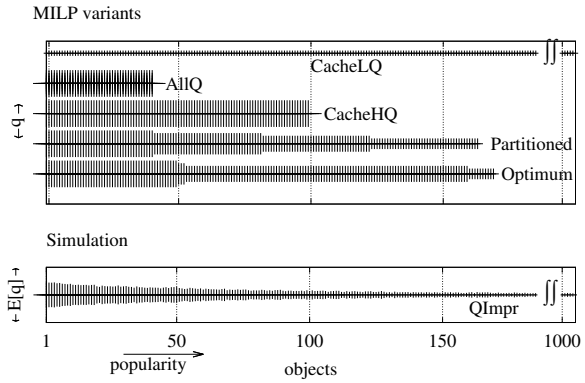



Figure 6. Single-AS scenario: Quality levels cached by centralized strategies (MILP variants) and distributed policy (simulation).

colors) are served by the cache and only low representations cross the upstream link. Indeed, it would not be beneficial to serve high quality objects through the upstream link, since the high bandwidth cost should be paid repeatedly, at each request. On the contrary, placing them in the cache permits to pay only once the cost in terms of memory, and to still repeatedly gather utility at each request.

Third, the load has an evident impact on the breakdown. At high load, both streams must carry lower quality representations. Indeed, in this case, the average quality of the upstream must be low, to fit the link capacity. At the same time, we need to reduce the number of transmissions on the upstream by intercepting more requests with cached copies. To do so, we need to cache a larger number of different videos and, since the cache space is limited, we need to store smaller copies of them, i.e. lower quality representations. This explains why, at high load, the quality of the cache-stream decreases.

Fourth, we observe the impact of cache size on the breakdown: as expected, when the cache size increases, its relative contribution to the overall utility increases as well. Yet, more interestingly, also the breakdown of the stored video quality changes as well: in particular, the larger the cache, the higher the quality, which is intuitive.

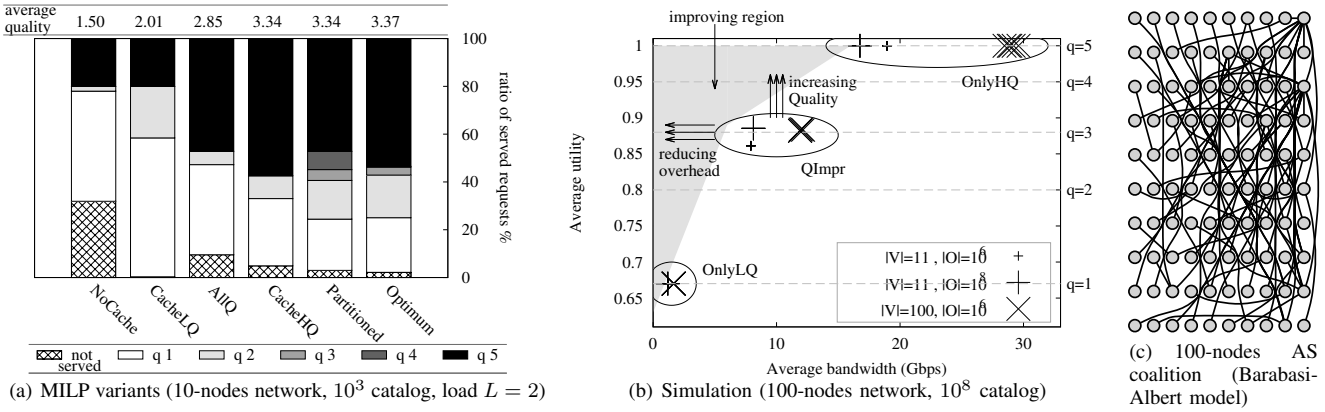
Finally, observe that the cache size has a side effect on the breakdown of the upstream video quality: indeed, the average quality increases for increasing cache size, which can be explained with the fact that the larger the cache, the larger the fraction of absorbed traffic. As a consequence, at any given load the upstream link has to serve less requests and can afford to do it at higher quality.

### *D. Performance bounds of online algorithms*

We next compare the performance of the five strategies discussed in Sec. III-D (viz., *NoCache, CacheLQ, CacheHQ, AllQ, Partitioned*), with the solution that maximizes the quality of experience perceived by network users (*Optimum*). Utility is reported in Fig. 5, whereas the structure of the solution is reported in Fig. 6, which depicts the quality level of each stored object under all strategies – including the distributed *QImpr* policy discussed in Sec. III-E.

Note that, when the network caches only low quality objects (*CacheLQ*), their small size permits to store a large number of them, intercepting a large fraction of the requests. This already provides robustness with respect to load, guaranteeing at least a minimum quality (the *CacheLQ* curve is above the $q = 1$ reference quality), that is not possible without cache. However, *CacheLQ* does not exhibit cache efficiency because higher quality objects, necessary to increase the average utility, can only be retrieved through the upstream link. Rigidly storing all quality representations (*AllQ*) further improves the performance but is still far from the optimum. Indeed, for each object we must waste cache space for all the representations, although only a subset of them will be actually served to users. This limits the number of different objects that can be actually cached. *CacheHQ* performance approaches to the *Optimum*, suggesting that storing few (due to their large size) popular objects at HQ already provides a notable payoff (due to the product of their popularity times their utility at high quality).

Yet, *Partitioned* performance is even closer to the *Optimum*: the root cause is that the quality representation selection is similar to the optimal one, as Fig. 6 shows. In particular, the optimal behavior in terms of overall utility is to store a number of objects at each quality, preferring to store more popular objects at higher quality, and *Partitioned* implements this behavior. This increases the overall cardinality of cached




(a) MILP variants (10-nodes network, $10^3$ catalog, load $L = 2$)

(b) Simulation (100-nodes network, $10^8$ catalog)

(c) 100-nodes AS coalition (Barabasi-Albert model)

Figure 7. Multi-AS scenarios: (a) optimal solution, (b) simulation results and (c) example large-scale topology.

content and assigns to each object the "right" quality, i.e. the one such that the cost in terms of occupied memory is compensated by the pay-off in terms of utility provided to the set of requests for it. The difference between *Partitioned* and *Optimum* is in the number of objects stored at each quality. While *Partitioned* constrains this number to be the same for each quality, *Optimum* does not incur this constraint and prefers, in this scenario, roughly two quality levels. So doing, the *Optimum* strategy caches more objects than *CacheHQ* (but less than *Partitioned*), a significant fraction of which is at lower quality than *CacheHQ* (but higher than *Partitioned*).

From the above observations, we infer that the quality at which each object must be cached should increase with its popularity. This is the observation we leveraged in the design of *QImpr*, which is shown in the last raw of Fig. 6. Notice that while solving the optimization problem returns exactly one object quality, in the simulation case the representation of an object stored in the cache varies over time, so that we report the average quality for an object sampled at 100 random times during the simulation. It can be seen that *QImpr* tends to store only the popular objects at high quality, thus approaching a solution that is structurally similar to *Partitioned* or *Optimal* strategies – which confirms the mechanism of improving the object quality at each new hit to pay off. Note that a fairness concern may arise, since popular content is served better than the rest. In any case, bandwidth is limited and it is impossible to serve all the content at high quality. Therefore, a network provider has two choices: i) being fair and lowering the quality of all the served videos or ii) differentiating based on popularity. While the former case is admissible, we have shown that the latter permits utility maximization, which is the target of this paper. On the other hand, a network provider may wish to provide always a quality above a certain threshold higher than LQ. We can easily model this by removing from the set $Q$ of the admissible levels the lowest ones.

### *E. Realistic and large scale topologies*

We now consider a multi-AS environment, where each AS operates a cache system with a storage space sufficient to cache 1/100 of the catalog at the highest quality. We start our analysis solving the MILP model for a coalition of 10 ASes and a $10^3$ objects catalog in Fig. 7-(a), and extend the analysis up to 100 ASes and $10^8$ objects in Fig. 7-(b) The multi-AS graphs are generated in accordance to the Barabasi-Albert model [4], which is considered to approximate the AS interconnection in Internet [19]. A compact illustration of the interconnection is in Fig. 7-(c) for a large scale topology of 100 nodes.

We only briefly comment MILP results, reported in Fig. 7-(a), to assess that they are coherent with the observations on the single-AS case. Specifically, we notice that while the average quality is very similar among *CacheHQ, Partitioned* and *Optimum*, however the fraction of content that is not served is largely different. In the case of *CacheHQ*, about 5% of the videos are not served, which is 2.3 times larger than the fraction of non-served videos in the *Optimum* case. In contrast, the *Partitioned* strategy limits to +30% the amount of additional videos not served with respect to *Optimum*.

While this fact does not appear in the perceptual model we used (where a non served content has a utility 0 and does not generate any penalty) nevertheless it can be argued that the impact of service denial can be much worse. Indeed, from loss aversion models commonly used in prospect theory [11], not receiving a video at expected quality $q$ generates a negative utility $-2u(q)$, which could be accounted for in the model. Yet, repeatedly receiving denial of requests could lead users to change ISPs on a long timescale, which can have disastrous consequences on the ISP business, for which limiting the fraction of non-served content is primordial.

A second important observation is that gains are structurally equivalent to what early shown in the single-AS case, and on which we based the design of our proposed distributed strategy (*QImpr*). Comforted by this observation, we relax the capacity constraint and carry out simulation of *QImpr* on large scale instances. Otherwise stated, while the *Optimum* operates at full capacity, we do not expect ASes to run their network at this capacity regime. Rather, ISPs will be interested in controlling their average (or peak) bandwidth on external links, which we assess by simulating scenarios with dynamic arrivals. Aiming at assessing the utility vs. bandwidth trade-off, we use two additional reference points where only low-quality (*OnlyLQ*), or high-quality objects exist in the system (*OnlyHQ*), and caches employ standard Least Recently Used (LRU) replacement. Requests are generated according to a Poisson process and results are collected in steady state over 10 runs for each scenario. The bandwidth utilization is computed

considering that, every time an object at quality $q$ crosses a link, it occupies a bandwidth $r^q$. The bandwidth in Fig. 7-(b) is averaged over time and over all the links of the network.

Note that the points in Fig. 7-(b) are well clustered, meaning that the performance of *OnlyLQ*, *OnlyHQ* and *QImpr* is coherent and our findings do not vary with the scale of the problem. Bounds on the utility vs. bandwidth trade-off are given by *OnlyLQ* and *OnlyHQ*: the former guarantees the minimum bandwidth utilization by only serving and caching objects at the lowest quality, while the latter provides maximum utility at the expense of high bandwidth utilization. *QImpr* nicely fits halfway these extremes, realizing a smooth tradeoff between bandwidth and quality.

The picture finally shades an area where the performance of interesting distributed algorithms lays: i.e., those that achieve a more convenient bandwidth-quality tradeoff. *QImpr* design can be ameliorated to move performance in the upper-left part of Fig. 7-(b) by (i) reducing the overhead (i.e., move left) and (ii) improving the utility (i.e., move up). As far as (i) *overhead* is concerned, recall that whenever a request hits a cached copy at quality $q$, the cache immediately triggers a request to improve the content quality to $q+1$. These cache-originated requests constitute an overhead, which could be limited by probabilistically reducing the rate at which they are issued – much as in probabilistic meta-caching. As far as (ii) *utility* is concerned, recall that the *Optimal* solution implicitly quantized the quality levels to a subset of all the available ones, which should be easy to implement.

Notice however that overhead reduction and utility maximization are conflicting goals, since e.g., slowing down the rate at which quality of content is improved from $q$ to $q+1$ by a given factor also implies that the amount of requests served at quality $q$ instead of $q+1$ grows by the same factor. While this observation affects only the transient but vanishes in the steady state, it can however be argued that it has practical relevance in real scenarios where popularity is time-varying and there is no steady state. Additionally, the choice of the best subset may depend on the utility, cache/upstream ratio, topology, request load, popularity skew, etc. which requires future work.

## V. Conclusions

To the best of our knowledge, this is the first paper that tackles the problem of optimal content distribution in cache-enabled networks, by explicitly taking into account multiple representations of the same object, each having a different utility perceived by users. This is a crucial aspect in video delivery, in which each video can be represented at different quality levels. The need for caching techniques that, apart from the general ones, are optimized for video traffic is enforced by the prevalence of this traffic on the other types and its inherent cacheability. We find the optimal caching solution that maximizes user utility and we contrast it against several candidate strategies along the user experience angle. We study the fundamental properties of the solution to infer important guidelines to optimize object-level caching in video delivery. We leverage these guidelines in designing a distributed solution that we benchmark via event-driven simulation. Our key findings suggest that (i) the quality at which each object should be cached is inversely related to its popularity, (ii) a balance between user perceived utility and bandwidth usage is possible by means of intelligent caching distributed policy of which *QImpr*, the one proposed in this paper, is an example.

However, *QImpr* does not allow ISPs to explicitly control the balance, so to reach a target network utilization. In our future work, we aim at (i) proposing a distributed solution that approaches a target optimal bandwidth-quality tradeoff, as well as (ii) performing a thorough sensitivity analysis on the topology of the coalition, investigating how cache content differentiates with respect to the node position in the network, due to the interaction and the filtering effect of neighbors.

## References


[1] Sharp Website. http://www.sharpusa.com/ces-2015-recap.aspx.

[2] Reddit Website. http://www.reddit.com/r/netflix/comments/2uzu1s/why_is_youtubes_adaptive_streaming_so_bad_and/.

[3] M. Badov, A. Seetharam, and J. Kurose. Congestion-Aware Caching and Search in Information-Centric Networks. In *ACM SIGCOMM ICN*, 2014.

[4] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[5] S. Borst, V. Gupta, and A. Walid. Distributed Caching Algorithms for Content Distribution Networks. In *IEEE INFOCOM*, 2010.

[6] G. Cofano, L. De Cicco, and S. Mascolo. A Control Architecture for Massive Adaptive Video Streaming Delivery. In *ACM VideoNext Workshop*, 2014.

[7] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo. ELASTIC: a Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *IEEE Packet Video Workshop (PV)*, 2013.

[8] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman. On the Complexity of Optimal Routing and Content Caching in Heterogeneous Networks. In *IEEE INFOCOM*, 2015.

[9] A. Finamore. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *ACM SIGCOMM IMC*, 2011.

[10] Y. Jin, Y. Wen, and C. Westphal. Towards Joint Resource Allocation and Routing to Optimize Video Distribution over Future Internet. *IFIP Networking*, 2015.

[11] D. Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[12] D. H. Lee, C. Dovrolis, and A. C. Begen. Caching in HTTP Adaptive Streaming: Friend or Foe? In *ACM NOSSDAV Workshop*, 2014.

[13] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj. Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network. *Elsevier Signal Processing: Image Communication*, 27(4):288–311, Apr. 2012.

[14] B. Niven-Jenkins, F. L. Faucheur, and N. Bitar. Content Distribution Network Interconnection (CDNI) Problem Statement. IETF RFC 6707, September 2012.

[15] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas. Video Delivery over Heterogeneous Cellular Networks: Optimizing Cost and Performance. *IEEE INFOCOM 2014*, 2014.

[16] Y. Sun, S. K. Fayaz, Y. Guo, V. Sekar, Y. Jin, M. A. Kaafar, and S. Uhlig. Trace-Driven Analysis of ICN Caching Algorithms on Video-on-Demand Workloads. In *CoNEXT*, 2014.

[17] H. Susanto, B. Kim, and B. Liu. User Experience Driven Multi-Layered Video Based Applications. In *IEEE ICCCN*, 2015.

[18] L. Toni, R. Aparicio-Pardo, G. Simon, A. Blanc, and P. Frossard. Optimal Set of Video Representations in Adaptive Streaming. In *ACM MMSys*, 2014.

[19] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie. Optimal Cache Allocation for Content-Centric Networking. *IEEE ICNP)*, 2013.

[20] S. Yun, D. Kim, X. Lu, and L. Qiu. Optimized Layered Integrated Video Encoding. *IEEE INFOCOM 2015*, 2015.

[21] 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*, 15(6):1431–1445, Oct. 2013.