Optimization of Content Caching in Content-Centric Network

Tuan-Minh Pham, Michel Minoux, Serge Fdida, Marcin Pilarski

Optimization of Content Caching in Content-Centric Networks

Tuan-Minh Pham*, Michel Minoux†, Serge Fdida†, Marcin Pilarski‡
*Faculty of Information Technology, Hanoi National University of Education, Vietnam
†LIP6 Laboratory, UPMC Sorbonne Universités, France
‡Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland

***Abstract*—Video on demand (VoD) systems currently use content delivery networks (CDN) to distribute content to users, whose performance and effectiveness depends on the architecture, the number and the geographical location of CDN nodes deployed by CDN providers or ISP(s) itself. Content-Centric Networking (CCN), with the benefits of caching and sharing content by every node in the network, suggests an alternative: a collaborative caching system exploiting the maximum capacity of infrastructure for the high performance of video delivery services. However, a CCN-based architecture to support efficient VoD delivery raises important questions about the optimal routing and caching strategies with constraints on the architecture and capacities of the system. We investigate models and algorithms for addressing these optimization problems. We study different solutions for the routing and caching optimization problems and compare the solutions produced with the optimal solution under various assumptions. We also contribute to an analysis of the caching investment under the competition among multiple interconnected ISPs. Our numerical results show the influence of throwing caching at the problem in different locations, on the system performance and its related cost.**

***Index Terms*—Optimal caching, CCN, integer programming, stochastic programming, non-cooperative game.**

I. Introduction

The Internet has evolved towards an amazing machinery to distribute content at scale. Nevertheless, the current service model might not be appropriate and novel architectures have been proposed based on various Information-Centric Networking architectures. Among them CCN (and more recently NDN) has been widely studied over the last few years. ICN decouples the sender from the receiver and provides caching capabilities in the network. The content is then possibly made available closer to the user(s), not only reducing network traffic and delivery delays, but also reducing Mean Opinion Score (MOS) variance and increasing QoE for End-User [18]. Besides the protocol issues raised by CCN, this solution triggers a potential for defining new roles and business opportunities for the various stakeholders, namely Internet Service Providers (ISP), Content Providers (CP) and Content Distribution Networks (CDN) [19].

In this paper, we explore solutions for optimizing the location of content. Our algorithm could be applied both in wired and wireless environments where caching content closer to the user is beneficial because of limited bandwidth or congestion risk. We consider the distribution of content, likely VoD, to end users connected either through their set top boxes or their wireless devices via a home gateway. We assume that the system is supported by the ability, for a provider to assess user's need thanks to a recommendation service alike those found in many professional VoD systems. Based on this information, we propose a strategy that optimizes the location of content towards users devices as well as routers with caching capabilities within the infrastructure of one ISP.

In the context of multiple ISPs, content demands can be serviced by a local ISP, a content provider, or other ISPs. In such contexts, ISPs are competitive with each others for minimizing their cost. Thus, we consider the following questions: Does an equilibrium exist and under which condition? At the equilibrium, what is the impact of caching investment on the utility of ISPs?

Our first contribution provides a feasible cache solution that minimizes the expectation of routing cost computed from the optimal routing solution. We formulate the optimization problem of CCN caching as a two-stage stochastic programming problem transformed to a deterministic multiscenario linear program. The main advance of the formulation is the deterministic evaluation of cost and constraints in a non-deterministic scenario involving the uncertainty of content demand. We investigate models and algorithms for addressing the routing and caching optimization problems. Several work has been devoted to the study of optimal caching in CCN [8], [24]. However, to the best of our knowledge, there were no attempt thus far to provide a formulation capturing the deterministic computation of cost and constraints while taking into account a set of content demand scenarios (i.e. a long time period, not just a snapshot of the system at a particular time).

We then extend our analysis of caching in a context of a single ISP to that in a multi-tiered hierarchy of interconnected ISPs. In a context of multiple ISPs, a decision of any ISP would have an impact on the strategies of the others while each ISP optimizes its decisions in an individual manner. We use a game-theoretical model to formulate the competition among multiple ISPs with regard to the impact of caching and congestion costs. Our numerical results infer that a local ISP cannot always receive a profit from its caching investment while it can always earn a profit from caching investment of other ISPs. The ISP can use our game-theoretical model to analyze its utility under competition with other ISPs by adapting the cost and revenue functions to various contexts.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III is devoted to a brief description of the system under consideration. We state formally the caching problem in Section IV. Our models and algorithms for routing and content location are presented in Section V. Section VI describes our game-theoretical model for analyzing content caching in a multi-tiered hierarchy of interconnected ISPs. Section VII presents our evaluation and discussion of the performance of our solution, and the impact of caching investment of ISPs under the competition. Section VIII concludes the paper and highlights future work.

II. Related Work

Information-Centric Networks are widely studied with solutions such as PSIRP [3], DONA [16] or NDN [2]. The CCN framework was first introduced by Van Jacobson and the PARC research group in [13], [1]. Various issues arising in CCNs have been considered such as content router issues [5], data transfer modelling [10] or chunk-level caching [11]. Content caching was strongly investigated in different contexts in the Internet. Li et. al. addressed the optimal placement of web proxies for networks with a tree topology [17]. Qiu et. al. proposed greedy algorithms to find the optimal location of web servers for real network topologies [21]. In [15], the authors proved that the optimization problem of object replication in CDNs is NP complete and proposed heuristics for finding near-optimal solutions. In [25], Yu et. al. studied the impact of the number of servers and their locations on the aggregate throughput and operating cost in CDNs. However, the above solutions as well as the ones designed for current multi-cache networks such as the Web and CDNs are not applicable to a CCN environment due to CCN's unique properties including: 1) content is located by name instead of location, and 2) every ICN node can cache and serve the requested content.

Several papers were published to study the problem of content caching in CCN for better performance and efficient resource utilization. [20], [23], [22], [12]. In [22], Rosensweig et. al. provide an approximate model for analyzing the performance of CCNs where contents are cached at each node along the path for delivering the requested contents to customers. Psaras et. al. consider the modelling and evaluation of caching policies based on Markov chains [20]. In [23], Rossi and Rossini propose a solution to the cache allocation problem for individual CCN routers by using centrality metrics such as betweenness, closeness and degree centralities. In [12], the authors use trace-driven simulations to evaluate the performance benefits achieved by CCNs. Recently, Araldo et. al. propose a cost-aware cache decision policy whose objective is the cost reduction, contrarily to the above studies that focus on caching efficiency [4]. Our work is different as it considers the joint problem of routing and caching in a CCN context where any node can cache and share content. So far, there has been little discussion about the joint problem of routing and caching [14], [8], [24]. In [24], the cache allocation problem for CCNs is formulated as a 0-1 maximization problem featuring the structure of a knapsack problem. However the model relies on several simplifying assumptions such as fixed routing and without bandwidth constraints. In [14], the authors study adaptive mechanisms to manage content replication and routing on a continuous basis. In [8], the authors address the optimization of content caching and routing in a hierarchical tree network. However, to the best of our knowledge, there exists no attempt to provide a formulation capturing the deterministic computation of cost and constraints while taking into account a set of content demand scenarios (i.e. a long time period, not just a snapshot of the system at a particular time). In addition, no research has been found that surveyed competition among ISPs in a general multi-tiered hierarchy of interconnected ISPs with regard to the impact of caching and congestion costs. The model discussed below is intended to overcome such limitations.

III. System Description

We consider a content delivery architecture involving a Content Provider (CP) and a set of user's set-top boxes (STBs) connected to the CP through a network path. The later is reduced to an intermediate router for simplification purposes and also because it is likely that in-network caching will mostly be beneficial at the edges, namely the intermediate router in our model. The system provides content delivery with quality of service constraints such as video-on-demand alike Netflix for a set of geographically dispersed users. Each user subscribing to such services is equipped with a STB that allows to cache content objects. An intermediate router between the CP and STBs can store other content objects depending on its storage capacity as well as the caching strategies of the system. The CP store all content objects that can be requested from users. Figure 1 illustrates the system including one root node that represents the CP, one intermediate node that is the intermediate router, and a set of STB nodes. A user requests content through its STB. Depending on the routing policy, a content request can be satisfied from any node including STBs, the intermediate node, or the root. Such an architecture optimizes the cost of content delivery by exploiting the capacity of content caching and content sharing among all nodes in the system.

The cost induced by the content delivery system is mostly the cost of transmission for moving content. Ideally, and as the last mile has limited resources (transmission and storage), we would like to opportunistically store the appropriate content as close as possible to the consumer or in a location that will benefit from the shared request of different users willing to consume the same content. A single ISP will benefit from serving users in the same neighborhood, or from one of its intermediate device, avoiding being charged for fetching the content from a competitor or directly from the Content Provider server.

We assume that the probability distribution representing the user preferences regarding content access is captured via a Zipf law [9], recognizing the diversity of content but also the existence of very popular ones. For this reason, it is possible that the system will face flash crowd when some extremely

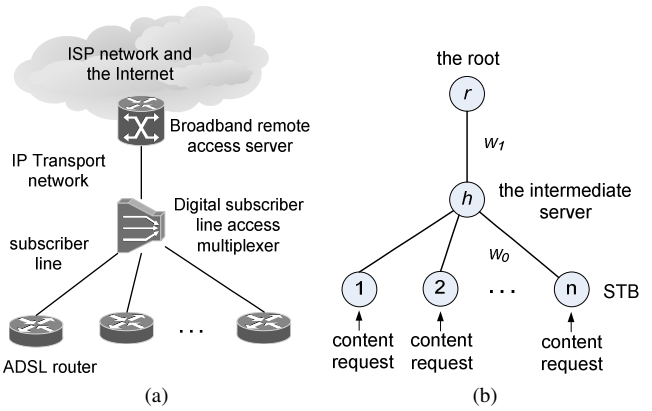

Fig. 1. Content delivery system: (a) An example of a physical VoD system (b) Model

popular content will be requested at similar times, increasing the congestion risk.

In order to optimize the content delivery cost under the system, we explore the strategy for caching content (placement) according to the user's needs, knowing that one can move content among nodes. Unlike other papers [14] we do not provide adaptive mechanisms to manage content replication and routing on a continuous basis but rather consider a different time scale where ISPs can optimize the placement of their content on a daily basis, exploiting information about usage statistics and preferences that are developed in current recommendation services. Therefore, we assume that ISPs can update their policy for content replication and placement at best times in order to take into account the evolution of the demand and the changing popularity of content.

IV. Problem Statement

The deployment of a content delivery system exploiting the maximum capacity of caching and sharing content among nodes requires to optimize the replication of content objects in order to minimize routing costs under bandwidth constraints. Specifically, for a given cache location, we need to solve the problem of optimal routing under bandwidth constraints of the system. The problem of optimal routing is to decide which node should serve a piece of content requested by a given other node so that all content requests of users are satisfied and the total transmission cost is minimized. The cost of a candidate solution for optimal cache location is the cost of optimal routing. Hence, the problem of optimal cache location is to find a solution of locating content objects that optimizes the transmission cost under the optimal routing strategy subject to constraints on bandwidth and storage capacity.

In order to formally state the problem, we introduce the following notation:

- h is the intermediate node; r is the root node.
- I is the set of STB nodes; $I_1 = I \cup \{h\}$, and $I_2 = I \cup \{h, r\}$.
- J is the set of content objects. Without loss of generality, we assume that content object j_1 is more popular than content object j_2 if $j_1 < j_2$.
- n is the total number of STB nodes; m is the total number of content objects.
- s_I and s_h are the number of content objects that can be stored at a STB and the intermediate node respectively.
- c_0 is the capacity of the uplink (i, h) for each $i \in I$. c_0 is a small integer, specifying the maximum number of content objects stored at i which can be uploaded to other customers during a given time period (e.g. the peak hour time period in a typical day). We do not specify a *downlink capacity* because we assume that in all realistic scenarios for demands, the downlink capacity is sufficient to download the content objects required by any customer i (either from r, from h, or from another customer i').
- $y = (y_{ij})$ ($i \in I_1$, $j \in J$) is a candidate solution of cache location, where node i stores content j if $y_{ij} = 1$, or not if $y_{ij} = 0$.
- $d = (d_{ij})$ ($i \in I$, $j \in J$) is content demand in a given scenario of demands where d_{ij} is a 0-1 random variable. If node i requests content j, $d_{ij} = 1$, otherwise $d_{ij} = 0$. The various random variables d_{ij} are assumed to be independent.
- w_1 and w_0 are associated transmission costs when a content object is transmitted by a link between r and h, and a link between h and $i \in I$ respectively. We assume $w_1 > w_0$ due to the fact that the connection between the intermediate server and the ISP network is long-distance and uses an expensive technology in data transmission.

For content object j required by customer $i \in I$, the routing cost of satisfying that content request by node $i' \in I_2$, denoted by $w_{ii'j}$, is

- 0 if content object j is available at i (i.e. $i' = i, y_{ij} = 1$),
- w_0 if it is downloaded from h,
- $2w_0$ if it is downloaded from another customer $i' \neq i$,
- $w_0 + w_1$ if it is downloaded from r.

For $i \in I$, $i' \in I_2$, and $j \in J$, the routing decision variable $x_{ii'j} \in \{0, 1\}$ denotes whether or not content j required by node i is delivered from node i'. The routing cost of a possible routing solution x for a scenario of content demand d and content caching y is

$$\varphi(x, y, d) = \sum_{i \in I} \sum_{i' \in I_2} \sum_{j \in J} w_{ii'j} x_{ii'j} d_{ij}.$$

Problem 1 (Routing Problem): Given cache location (y_{ij}) where $i \in I_1$ and $j \in J$, and a scenario of content demand (d_{ij}) where $i \in I$ and $j \in J$, find a routing solution $(x_{ii'j})$, where $i \in I$, $i' \in I_2$ and $j \in J$, satisfying all the requirements of the customers in order to minimize routing cost $\varphi(x, y, d)$ subject to constraints on uplink bandwidth.

Suppose that we know the probability distribution of the possible scenarios of demands. Specifically, for each customer i, content j is required with a given probability $p_{ij} \geq 0$ (i.e. $P\{d_{ij} = 1\} = p_{ij}$, $P\{d_{ij} = 0\} = 1 - p_{ij}$). For a given probability distribution of demands, we denote $\psi(y)$ the expectation of routing cost with respect to a feasible caching location y. Then, the optimal caching problem is defined as follows.

Problem 2 (Caching Problem): Given the probability distribution of content demand, find $y = (y_{ij})$, where $i \in I_1$ and $j \in J$, in order to minimize $\psi(y)$ subject to constraints on storage capacity.

Next section will develop our solution for solving the above optimization problems.

V. Algorithms

A. Linear Programming Model

We consider the optimization problem of caching involving the uncertainty of content demand. Our goal is to find a feasible solution that minimizes the expectation of routing cost. Hence, the caching problem is a two-stage stochastic programming problem. Let x^* be the optimal solution of the routing problem (i.e. the second-stage problem) for a scenario of demand d, and $\varphi^*(y,d) = \varphi(x^*, y, d)$ be the optimal routing cost. Then, the expectation of routing cost for a candidate of caching solution is $\psi(y) = E[\varphi^*(y,d)]$. The two-stage formulation of the stochastic programming model (P_1) for the optimal caching problem is given by:

$$\begin{aligned}
\text{Minimize} \quad & \psi(y) = E[\varphi^*(y,d)] \\
\text{Subject to:} \quad & \sum_{j\in J} y_{ij} \leqslant s_I \quad \forall i \in I \\
& \sum_{j\in J} y_{hj} \leqslant s_h \\
& y_{ij} \in \{0,1\} \quad \forall i \in I_1, j \in J
\end{aligned}$$

where $y = (y_{ij})$ such that $y_{ij} \in \{0,1\}$ for $i \in I_1$ and $j \in J$.

In order to solve efficiently the above two-stage stochastic program, we transform it into a deterministic multiscenario linear program. We consider s scenarios of content demand generated from the popularity distribution of content requests. Each scenario is obtained by drawing independently a value of each variable d_{ij} according to the probability distribution $(p_{ij}, 1-p_{ij})$. Let $\pi^k = (\pi^k_{ij})$ be scenario k of content demand where $k \in K$ and $K = \{1, 2, \ldots, s\}$ is the set of s scenarios. In scenario k, if content j is requested by customer i, $\pi^k_{ij} = 1$, otherwise $\pi^k_{ij} = 0$. We denote $x^k = (x^k_{ii'j})$ a possible routing solution for content location (y_{ij}) and scenario k of content demand. The equivalent linear programming model (P_2) for the stochastic program (P_1) of the caching problem is given by:

$$\text{Minimize} \quad \frac{1}{s}\sum_{k\in K} \varphi(x^k, y, \pi^k) \tag{1}$$

$$\text{Subject to:} \quad \sum_{i\in I\setminus\{i'\}} \sum_{j\in J} x^k_{ii'j} \leqslant c_0 \quad \forall i' \in I, k \in K \tag{2}$$

$$\sum_{i'\in I_2} x^k_{ii'j} = \pi^k_{ij} \quad \forall i \in I, j \in J, k \in K \tag{3}$$

$$\sum_{j\in J} y_{ij} \leqslant s_I \quad \forall i \in I \tag{4}$$

$$\sum_{j\in J} y_{hj} \leqslant s_h \tag{5}$$

$$x^k_{ii'j} \leqslant y_{i'j} \quad \forall i \in I, i' \in I_2, j \in J, k \in K \tag{6}$$

$$x^k_{ii'j} \in \{0,1\} \quad \forall i \in I, i' \in I_2, j \in J, k \in K \tag{7}$$

$$y_{ij} \in \{0,1\} \quad \forall i \in I_1, j \in J \tag{8}$$

In linear programming model (P_2), conditions (4) and (5) are storage capacity constraints. For each scenario k of content demand, a feasible routing solution has to satisfy constraints on uplink capacity (2), the content availability at a sending node (6), the fulfilment of all content requests (3).

Unfortunately, data placement problems are NP-hard [6]. This implies that no polynomial time algorithm is known that solves exactly any of these problems. It is time consuming to solve the huge linear programming model (P_2) when the size of the program is large (i.e. thousands of content objects, hundreds of nodes, and hundreds of scenarios). Hence, in the sequel we propose heuristic that provide a solution close to the optimal with a reduced computation time.

B. Heuristics for Optimal Routing

We first consider the routing subproblem and propose a heuristic algorithm, namely Closest and Least Busy Node First Routing (CLBR), which provides a near-optimal routing solution with linear time complexity. The main ideas underlying this heuristic procedure are that the cost of providing content from a node that is closer to the user is cheaper, and contents whose popularity is low are rarely cached in a STB. The algorithm uses a priority list of STBs which suggests which STB should provide a content object when several STBs hold the content object. The priority of STB i is

$$\text{pri}(i) = \frac{1}{\sum_{j\in J} p_{ij} d_{ij}}. \tag{9}$$

The detail of all steps is summarized in Algorithm 1.

Proposition 1 shows two scenarios in which the heuristic algorithm provides the optimal solution to the routing problem.

Proposition 1: The CLBR algorithm provides the optimal routing cost if the uplink capacity is unlimited or if no STB is willing to share content.

Proof: Suppose the optimal solution is not the one produced by the algorithm. It means that one of policies 1-2 is suboptimal. Suppose policy 1 is suboptimal, it means that

Algorithm 1 Closest and Least Busy Node First Routing (CLBR)

When a user requests a content object through its STB, the request is satisfied by the following policies:

1) The local STB serves the request if the content object is available in its cache.
2) Otherwise, the intermediate node serves the request if the content object is available in its cache.
3) Otherwise, the node serving the request is the STB that has cached the content object and has the highest priority.
4) Otherwise, the root serves the request.

there exist $j \in J$ and $i \in I$ such that $y_{ij} = 1$, and either $x_{ii'j} = 1$, or $x_{ihj} = 1$, or $x_{irj} = 1$ where $i' \in I$ and $i' \neq i$ in the optimal solution. Suppose $x_{ii'j} = 1$, we build a feasible solution by changing $x_{ii'j}$ to 0 and x_{iij} to 1. Similarly, we can build a feasible solution when $x_{ihj} = 1$, or $x_{irj} = 1$. That feasible solution provides a lower cost, which contradicts the assumption. So, the optimal solution follows policy 1. Using similar arguments, we prove that the optimal solution follows policy 2, which demonstrates Proposition 1 when no STB is willing to share content.

Note that any feasible routing solution can be built by changing x_{irj} to 0 and $x_{ii'j}$ to 1 when content sharing is considered and the uplink capacity is unlimited. Using similar arguments used in the case of no sharing support, we prove that the algorithm provides the optimal solution when the uplink capacity is unlimited. ■

C. Heuristics for Optimal Caching

We propose a heuristic algorithm for finding an efficient solution to the caching problem, based on the local popularity of content requests. This heuristic is referred to as LPC. More specifically, the content object that a user requests with high probability will be stored locally in its STB. The entire process is presented in Algorithm 2.

Algorithm 2 High Local Popularity First Caching (LPC)

The policies of storing a content object locally are as follows:

1) For any STB, select the maximum number of content objects by descending priority of the content popularity.
2) For the intermediate server, select content objects that have not been cached in any STB by descending priority of the content popularity.

Proposition 2 shows a situation in which the LPC heuristic provides the optimal solution for the caching problem.

Proposition 2: The LPC algorithm provides the optimal solution if the intermediate node does not cache content and no STB is willing to share content.

Proof: Suppose the optimal solution is not the one produced by algorithm 2. It means that there exist $j_1, j_2 \in J$ and $i_k \in I$ in the optimal solution $y = (y_{ij})$ such that the request popularity of content j_1 is greater than the one of content j_2, $y_{i_k j_1} = 0$, and $y_{i_k j_2} = 1$. We build a feasible solution y' from y by changing $y_{i_k j_1}$ to 1, and $y_{i_k j_2}$ to 0. Let $\psi^s(y)$ be the total cost of s scenarios of content demands for the content location y. We denote d^s_{ij} the number of requests for content j required through STB i in these scenarios. Since no STB is willing to share content, following Proposition 1, the optimal routing cost can be computed by the CLBR algorithm. In addition, the intermediate node does not hold any content object. So, we have

$$\psi^s(y) = d^s_{i_k j_1}(w_0 + w_1) + \psi_0$$
$$\psi^s(y') = d^s_{i_k j_2}(w_0 + w_1) + \psi_0$$

where

$$\psi_0 = \sum_{i' \in I_2} \sum_{j \in J \setminus \{j_1, j_2\}} w_{i_k i' j} x_{i_k i' j} d^s_{i_k j} + \sum_{i \in I \setminus \{i_k\}} \sum_{i' \in I_2} \sum_{j \in J} w_{i i' j} x_{i i' j} d^s_{ij}$$

Since the request popularity of content j_1 is greater than that of content j_2, we have $d^s_{i_k j_1} > d^s_{i_k j_2}$. So, $\psi^s(y') < \psi^s(y)$. It follows that the feasible solution produces a lower cost, which contradicts the assumption. Hence, the algorithm delivers the optimal solution. ■

When the content demand of STBs is homogeneous (i.e. $p_{ij} = p_{i'j} = p_j$ for $\forall i' \neq i$), the cost of s scenarios of content demand is given by

$$\psi^s(y) = \sum_{j=s_I+1}^{s_I+s_h} n d_j w_0 + \sum_{j=s_I+s_h+1}^{m} n d_j (w_0 + w_1) \quad (10)$$

where we recall that m denotes the number of content objects and d_j is the total requests for content object j in all scenarios required by a STB. Assume that the popularity of a content request follows a Zipf distribution, then the request probability of content item of rank j is given by

$$p_j = \frac{1}{j^\alpha \sum_{z=1}^{m} \frac{1}{z^\alpha}}$$

where α is the value of the Zipf's exponent depending on the type of content. When very few popular content objects exist (i.e. $p_j \to 0$, $d_j \to 0$ as j is large), $\psi^s(y)$ is small. It follows that the result provided by the algorithm is close to the optimal when the value of the Zipf parameter for the popularity of content request is high. When the value of the Zipf parameter is small, we propose to use the adaptive popularity algorithm (APC) that adjusts the number of content objects stored locally to the content popularity. The basic idea of the APC algorithm is that the number of content objects stored locally is in proportion to the popularity of content request. For example, if the popularity of content request follows the Zipf distribution, the number of content object j stored locally is given by

$$s_j = \frac{n s_I + s_h}{\sum_{j=1}^{m} \prod_{k=j+1}^{m} k^\alpha} \prod_{k=j+1}^{m} k^\alpha.$$

Algorithm 3 Adaptive Popularity Caching (APC)

1) For content object $j \in J$, compute the number of copies s_j stored locally in all STBs based on the popularity of content requests.
2) For $j = 1$ to m
   a) Store a copy of content object j in the intermediate server if the number of copies is less than n.
   b) Store a copy of content object j in STB i by descending priority of p_{ij} until the number of copies reaches s_j.

The detail of all steps is presented in Algorithm 3.

Proposition 3 compares the results provided by the two algorithms.

Proposition 3: Suppose the content demand of STBs is homogeneous, the uplink capacity is unlimited, the intermediate server does not store any content object, the total storage capacity of all STBs is larger than or equal to m, and the popularity of a content object is uniform, the APC algorithm provides a better result than the LPC algorithm if

$$\frac{s_I}{m} < \frac{q-1}{q+1} \tag{11}$$

where

$$q = \frac{w_1}{w_0}$$

Proof: Since the uplink capacity is unlimited, the optimal routing can be computed by the CLBR algorithm. Because the content demand of STBs is homogeneous, the intermediate server does not store any content object, and the popularity of a content object is uniform, from (10), the cost of the solution provided by the LPC algorithm is given by

$$\psi^s_{LPC}(y) = n\,(m - s_I)\,d_j\,(1+q)\,w_0.$$

When the content demand of STBs is homogeneous, the uplink capacity is unlimited, the intermediate server does not store any content object, the total storage capacity of all STBs is larger than or equal to m, and the popularity of a content object is uniform, by applying the CLBR algorithm for routing, the cost of the solution provided by the APC algorithm for the caching problem is

$$\psi^s_{APC}(y) = nmd_j 2w_0 - ns_I d_j 2w_0.$$

We have

$$\psi^s_{LPC}(y) - \psi^s_{APC}(y) > np_j d_j w_0\,[(1+q)\,(m - s_I) - 2m]$$

So, $\psi^s_{LPC}(y) - \psi^s_{APC}(y) > 0$ if condition (11) is satisfied, which proves the claim. ■

VI. Competition among Multiple ISPs

We study the impact of caching under the competition among multiple ISPs (Fig. 2). Let $\mathbb{N}$ be a set of content providers. $\mathbb{H}$ is a set of local ISPs. $\mathbb{G}$ is a set of regional ISPs. We denote by $C^a_{hg}(\kappa_{hg})$ the cost function of caching investment of local ISP $h \in \mathbb{H}$ whose provider ISP is regional

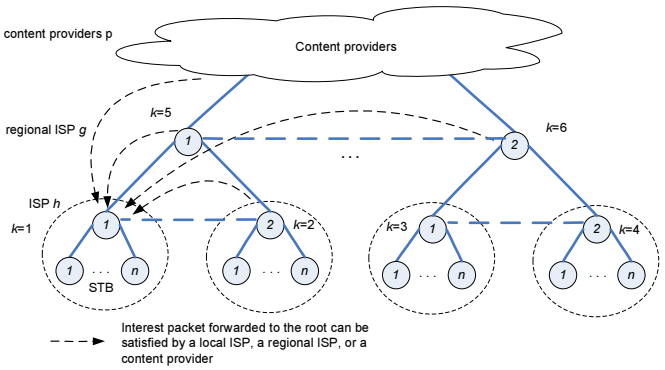

Fig. 2. Caching under competition among multiple ISPs

ISP $g \in \mathbb{G}$, where $\kappa_{hg} \in [0,1]$ is the caching factor. We will refer to local ISP h connected to regional ISP g as ISP (h,g). ISP (h,g) does not invest in caching if $\kappa_{hg} = 0$, and it can cache all content items if $\kappa_{hg} = 1$. Let ${}^k q_{hg}$ be a number of content demands that are requested by STBs of ISP (h,g), and satisfied by node k ($k \in \mathbb{K}$, $\mathbb{K} = \{p\} \cup \mathbb{G} \cup \mathbb{H} \setminus \{h\}$). ${}^k p_{hg}$ is the transmission cost when one of ${}^k q_{hg}$ content demands is fulfilled. $\widehat{p}_{hg}$ is the vector of all transmission cost of node h of g.

Suppose that the number of content demand ${}^k q_{hg}$ is increasing in caching factor κ_k and is decreasing in transmission cost ${}^k p_{hg}$. We consider a function of content demand as follows

$${}^k q_{hg} = D \frac{1}{e^{1-\kappa_k}} {}^k p_{hg}{}^{-\beta} \sum_{k' \neq k} {}^{k'} p_{hg} \sum_{h' \neq h, g' \neq g} {}^k p_{h'g'} \tag{12}$$

where β is a constant representing the sensitivity effects of demands on the cost of delivering a content item from k to ISP (h,g).

Let $c^k_{hg}({}^k q_{hg})$ be the congestion cost of a path from ISP (h,g) to node k, which is a function of the total content demands on the path. The congestion cost of a path increases if the total content demands on the path increases. Consider a linear function of congestion cost

$$c^k_{hg}\big({}^k q_{hg}\big) = {}^k q_{hg}\,\alpha \tag{13}$$

where congestion factor α is a constant.

We denote by $C^o_{hg}(\widehat{p}_{hg})$ the cost when all content demands requested by node h of g are satisfied. $C^o_{hg}(\widehat{p}_{hg})$ is the sum of the total congestion cost and the total transmission cost. Specifically, we have

$$C^o_{hg}(\widehat{p}_{hg}) = \sum_{k \in \mathbb{K}} c^k_{hg}\big({}^k q_{hg}\big) + \sum_{k \in \mathbb{K}} {}^k q_{hg}\,{}^k p_{hg}. \tag{14}$$

Let $\psi_{hg}(\widehat{p}_{hg})$ be the revenue that ISP (h,g) receives when servicing all content demands from its users. Suppose that the revenue responded to content request is linear

$$\psi_{hg}(\widehat{p}_{hg}) = \tau \sum_{k \in \mathbb{K}} {}^k q_{hg} \tag{15}$$

where τ is the revenue of the ISP when it fulfills one content demand.

The utility of local ISP h of regional ISP g equals the revenue minus costs. We have

$$U_{hg}(\widehat{p}_{hg}) = \psi_{hg}(\widehat{p}_{hg}) - C^{o}_{hg}(\widehat{p}_{hg}) - C^{a}_{hg}(\kappa_{hg}). \quad (16)$$

Consider a noncooperative game played by all local ISPs adjusting their prices, to maximize their utility. The strategy $\widehat{p}^{*} = \left\{ {}^{k}\widehat{p}_{hg} : k \in \mathbb{K}, h \in \mathbb{H}, g \in \mathbb{G} \right\}$ where $\widehat{p}^{*}_{hg} = \left\{ {}^{k}\widehat{p}^{*}_{hg} \right\}$ constitutes an equilibrium if $\widehat{p}^{*}$ solves the following optimization problems for all players (h, g) (i.e. local ISP h of regional ISP g)

$$\max_{\widehat{p}_{hg}} U_{hg}\left(\widehat{p}_{hg}, \widehat{p}^{*} \backslash \widehat{p}^{*}_{hg}\right) \quad (17)$$

Proposition 4 shows a condition under which a Nash equilibrium exists. Details of the proof for the proposition is given in A.

Proposition 4: There exists a pure Nash equilibrium in the competition between ISPs if

$${}^{k}p_{hg} \in \left[0, \frac{\beta + 2}{\beta\left(\tau - \alpha\right)}\right]. \quad (18)$$

Proposition 5 describes a pure Nash equilibrium where the decision of the ISPs on their prices and the caching factor are independent. It infers that a possible equilibrium under competition in ICNs is a state where all ISPs join together to exchange content and create a federated model (i.e. all ISPs share their cache and agree on a price).

Proposition 5: There is an equilibrium under which the caching factor has no effect on the decision of the ISPs:

$${}^{k}\widehat{p}^{*}_{hg} = \frac{\beta + 1}{\beta\left(\tau - \alpha\right)}. \quad (19)$$

Proof: We differentiate U_{hg} w.r.t ${}^{k}p_{hg}$, $\partial U_{hg} / \partial\, {}^{k}p_{hg}$, as computed in Appendix A. ${}^{k}\widehat{p}^{*}_{hg} = \frac{\beta+1}{\beta(\tau-\alpha)}$ is an equilibrium in the game. Indeed, substituting ${}^{k}\widehat{p}^{*}_{hg} = \frac{\beta+1}{\beta(\tau-\alpha)}$, we find $\frac{\partial U_{hg}}{\partial\, {}^{k}p_{hg}} = 0$. ■

VII. Evaluation

In this section, we evaluate the algorithm performance and impact of several parameters on the cost of the content delivery system in the context of one ISP. We then investigate the impact of caching investment of ISPs on their utility under the competition among multiple ISPs.

We first evaluate the LPC and APC algorithms under a practical setting. A video delivery network in practice uses ADSL/VDSL in rural and suburbia or DOCSIS/FTTH in urban areas as a data communications technology between end-users and an intermediate server. In both cases we do propose aggregation points like DSLAM (i.e. a digital subscriber line access multiplexer), HFC (i.e. a hybrid fiber-coaxial) or OLT (i.e. Optical Line Termination) with caching capacity. In the network modeled in this paper, the number of users managed by a DSLAM is 200 users on average with median about 400, however most DSLAMs are stack into tree-like structure due to the geographical arrangement of underlying network topology. In summary because of DSLAM stacking principle

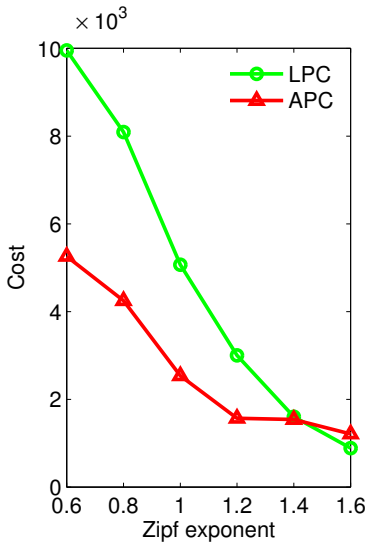

Fig. 3. Comparison between the LPC algorithm and the APC algorithm

the number of connected end-users into a single access router may be estimated to be on average about 2000. The video catalog of a content provider is composed of several thousand titles (e.g. Netflix's catalog contains approximately 18,000 active titles [7] and [26]). We compare the LPC heuristic and the APC heuristic under a scenario composed of 1000 STBs, and 10,000 content objects. We assume that a STB can store up to 5 content objects and the intermediate server can store up to 50 content objects. The uplink capacity between a STB and the intermediate server is $c_0 = 2$ content objects. The transmission cost between the intermediate server and the root, and the one between a STB and the intermediate server are set equal to $w_1 = 9$ and $w_0 = 1$ respectively. Using the above setting, we evaluate the heuristic algorithms under 100 scenarios of content demands when the content popularity follows the Zipf distribution with the exponent varying from 0.6 to 1.6. We observe the experimental results in Fig. 3. The APC algorithm provides a better result when the Zipf's exponent of the content popularity distribution is less than 1.4 whilst the LPC algorithm provides performs better for higher values of the Zipf's exponent.

Second, we study the impact of the uplink capacity on the routing cost. In our evaluation, the content popularity follows the Zipf distribution with the exponent $\alpha = 1.2$. The uplink capacity varies between 0 and 5 and other parameters are similar to those of the setting of the first evaluation. In Fig. 4, we observe that the cost significantly decreases when the system offers more sharing opportunities up to a point when the cost reduces slowly. Indeed, it is observed that the costs provided by APC and LPC respectively decreases by 70% and 25% by obtaining content from neighbor STBs when the uplink capacity changes from 0 to 1. The result suggests that a network provider only needs a small uplink bandwidth for improving the performance of its content delivery system, which is especially important for a network provider using a

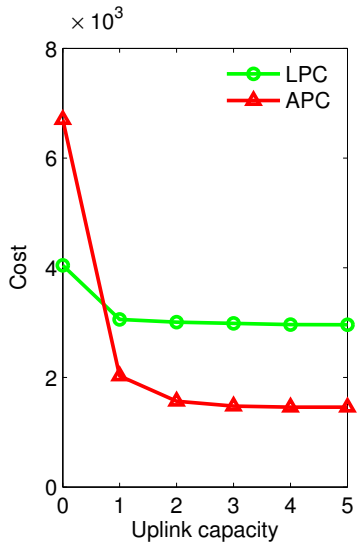

Fig. 4. Impact of uplink capacity

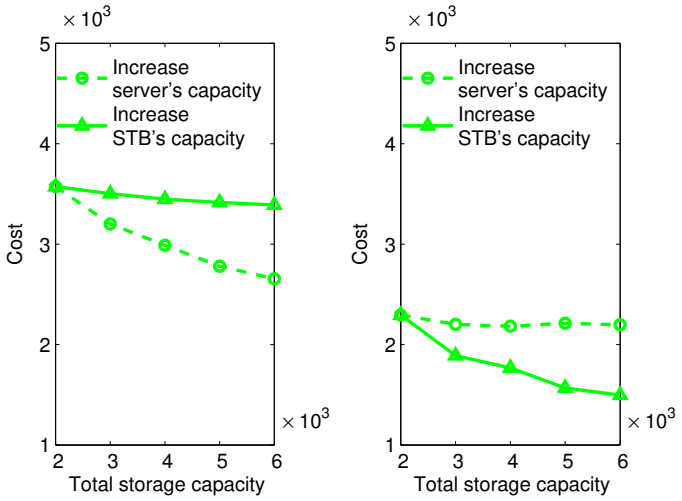

Fig. 5. Routing cost for LPC when distributing storage capacity between the server and STBs: (a) $\alpha = 0.8$, (b) $\alpha = 1.2$

TABLE I
PARAMETER SETTINGS FOR EVALUATING THE IMPACT OF ADDING STORAGE CAPACITY

	(Server's capacity, STB's capacity, total capacity)
Increase server's capacity	(1000, 1, 2000), (2000, 1, 3000), (3000, 1, 4000), (4000, 1, 5000), (5000, 1, 6000)
Increase STB's capacity	(1000, 1, 2000), (1000, 2, 3000), (1000, 3, 4000), (1000, 4, 5000), (1000, 5, 6000)

popular ADSL technology whose uplink bandwidth is limited.

Third, we evaluate the impact of distributing a fixed storage capacity between the intermediate server and the STBs. We consider a scenario where $n = 1000$, $m = 10,000$, $w_1 = 3$, $w_0 = 2$, and $c_0 = 2$ under 100 scenarios of content demands. We vary the storage capacity of the intermediate server or the STBs while keeping their total storage capacity fixed. Table I presents the parameter settings for the storage capacity in our evaluation. Figure 5 compares the routing cost of LPC in the case of adding storage capacity to the intermediate server (lines with circle markers) to the one in the case of adding storage capacity to the STBs (lines with upward-pointing triangle markers). Figure 5(a)-(b) shows the results when Zipf's exponent of the content popularity distribution is $\alpha = 0.8$, and $\alpha = 1.2$, respectively. We observe that adding storage capacity to the intermediate server is more valuable than adding storage capacity to the STBs when Zipf's exponent is low (i.e. $\alpha = 0.8$), but the result is opposite when Zipf's exponent is high (i.e. $\alpha = 1.2$). This implies that the content popularity has a major impact on the caching allocation policies.

For the purpose of comparison with optimal results, we now consider a limited size scenario composed of 10 STBs, 150 content objects, and 500 scenarios of content demands.

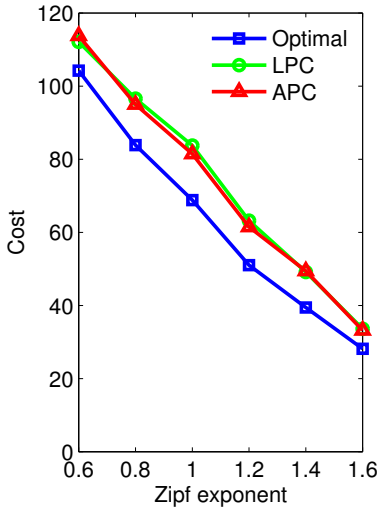

Fig. 6. Comparison between the heuristic algorithms and the optimal solution

In our evaluation, a STB can store one content object and the intermediate server can store up to 5 content objects. The uplink capacity between a STB and the intermediate server is 5 content objects. The transmission cost between the intermediate server and the root, and the one between a STB and the intermediate server are set equal to $w_1 = 10$ and $w_0 = 1$ respectively. We use the IBM ILOG CPLEX Optimizer to solve the linear programming model (P_2) in order to obtain optimal results. Figure 6 shows the results. We observe that the cost improves in the network where there are a few popular content objects, the results provided by the LPC and APC algorithms are close to the optimal result especially when the Zipf's exponent is high. In the figure, they are approximately 8 percent higher than the optimal result.

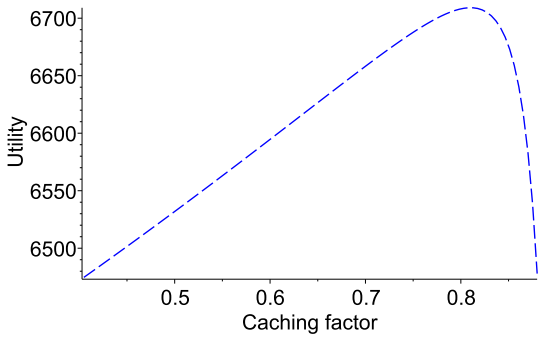

Fig. 7. Utility of local ISP 1 of regional ISP 1 at equilibrium when local ISP 1 of regional ISP 1 increases its caching investment

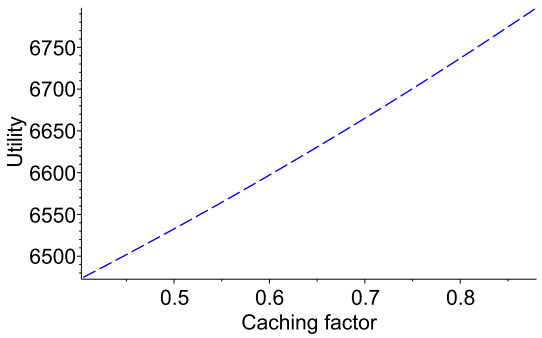

Fig. 8. Utility of local ISP 1 of regional ISP 2 at equilibrium when local ISP 1 of regional ISP 1 increases its caching investment

Finally, we study the impact of caching in a context of multiple ISPs including two regional ISPs and two local ISPs connected to each regional ISP. We consider the following function of caching cost

$$C^a_{hg}(\kappa_{hg}) = c_1 {c_2}^{\frac{c_3}{1-\kappa_{hg}}} - c_1 {c_2}^{c_3}$$

where c_1, c_2 and c_3 are constants. The value of the function tends to infinite as $\kappa \to 1$, and tends to zero as $\kappa \to 0$. These properties of the caching cost function agree with the fact that it is mostly impossible for the ISP to cache all content objects in the Internet and the ISP does not pay a caching cost if he does not invest in a caching system. For illustration purposes, the parameters of the caching cost function are given by $c_1 = c_3 = 1$ and $c_2 = 2$. We set the parameters of the caching cost function (12) to $D = 100$ and $\beta = 2$. The parameters of the congestion cost function (13) is $\alpha = 1$. The revenue of the ISP when fulfill one content demand in the revenue function (15) is $\tau = 3$. The caching factor κ of local ISP 1 of regional ISP 1 varies between 0.4 and 0.9 while the caching factor of other ISPs is 0.4. Fig. 7 and Fig. 8 plot the utility of local ISP 1 of regional ISP 1 and local ISP 1 of regional ISP 2 respectively. Fig. 7 shows that the utility of a local ISP increases until its caching investment reaches a threshold, and then it drops sharply. The results infer that it is not profitable for a local ISP to have a huge investment in caching in order to cache both content items with high popularity and those with low popularity. Fig. 8 shows that the utility of other local ISPs increases when one local ISP invests in caching. It occurs because a local ISP can provide a better service to its customers when it receives data packet from a neighbor ISP rather than a content provider that is far from its location.

The above results demonstrate that we have designed an efficient and tractable solution for routing and caching strategies in a CCN-based architecture providing VoD services. The solution can be deployed using a recommendation service that extrapolates user's interest as this exist in many operational platforms today. The ISP can use this solution at best times, facing a change in the demand and/or to benefit from opportunities in resource availabilities.

VIII. Conclusion

Our work introduces an architecture of VoD system on top of content-centric networking and addresses a joint optimization problem of content routing and content caching in the system. We believe such an architecture is highly beneficial in content delivery services as illustrated by our evaluation, which shows a significant improvement of performance with small investment in storage and bandwidth for content sharing. Our proposed heuristic algorithms for optimizing content routing and caching in the system were evaluated in both theoretical analysis and implementation, providing a practical solution to this problem. Our work also contributes to an analysis of caching investment in a context of multiple interconnected ISPs. The ISP can use our game-theoretical model to analyze its utility under competition with other ISPs by adapting the cost and revenue functions to various contexts. In future work, it would be of interest to consider storage costs in the objective function of the optimization problem, and to investigate their impact on the solutions obtained.

Appendix A
Proof of Proposition 4

From (12), (13), (14), (15), (16), we have

$$\begin{aligned}
U_{hg} &= \tau \sum_{k\in\mathbb{K}} {}^{k}q_{hg} - \alpha \sum_{k\in\mathbb{K}} {}^{k}q_{hg} - \sum_{k\in\mathbb{K}} {}^{k}q_{hg}\,{}^{k}p_{hg} - C^a_{hg}(\kappa_{hg}) \\
&= \tau D \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h, g'\neq g} {}^{k}p_{h'g'} \\
&\quad - \alpha D \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h, g'\neq g} {}^{k}p_{h'g'} \\
&\quad - D \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{1-\beta}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h, g'\neq g} {}^{k}p_{h'g'} \\
&\quad - C^a_{hg}(\kappa_{hg})
\end{aligned}$$

By differentiating U_{hg} w.r.t ${}^{k}p_{hg}$, we obtain

$$\frac{\partial U_{hg}}{\partial\, {}^{k}p_{hg}} = -\tau\beta D \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta-1}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h,k'\neq k} {}^{k}p_{h'g'}$$
$$+ \alpha\beta D \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta-1}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h,k'\neq k} {}^{k}p_{h'g'}$$
$$+ (\beta+1)\, D \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta-2}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h,k'\neq k} {}^{k}p_{h'g'}$$

$$\frac{\partial^2 U_{hg}}{\left(\partial\, {}^{k}p_{hg}\right)^2} = \beta\,(\beta+1)\,\tau D$$
$$\times \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta-2}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h,k'\neq k} {}^{k}p_{h'g'}$$
$$- \beta\,(\beta+1)\,\alpha D$$
$$\times \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta-2}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h,k'\neq k} {}^{k}p_{h'g'}$$
$$- (\beta+1)\,(\beta+2)\,D$$
$$\times \sum_{k\in\mathbb{K}} \frac{\left({}^{k}p_{hg}\right)^{-\beta-3}}{e^{1-\kappa_k}} \sum_{k'\neq k} {}^{k'}p_{hg} \sum_{h'\neq h,k'\neq k} {}^{k}p_{h'g'}$$

If ${}^{k}p_{hg} \in \left[0, \frac{\beta+2}{\beta(\tau-\alpha)}\right]$, we have $\partial^2 U_{hg} / \left(\partial\, {}^{k}p_{hg}\right)^2 < 0$. Thus, U_{hg} is concave. Since a concave function is quasiconcave, U_{hg} is quasiconcave. We have the sets of actions of all ISPs are nonempty compact convex subsets of a Euclidian space, and the utility function U_{hg} of the ISPs are continuous and quasi-concave on their set of actions. Hence, there exists a pure Nash equilibrium.

ACKNOWLEDGMENTS

The work presented in this paper has been carried out at LIP6 (http://www.lip6.fr) and LINCS (http://www.lincs.fr). The first author was partially supported by MOET Grant B2016-SPH-17.

REFERENCES

[1] CCNx project. Website http://www.ccnx.org.
[2] NDN project. Website http://named-data.net.
[3] Publish-subscribe internet routing paradigm (PSIRP) project. Website http://www.psirp.org.
[4] A. Araldo, D. Rossi, and F. Martignon. Cost-aware caching: Caching more (costly items) for less (isps operational expenditures). *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1316–1330, May 2016.
[5] S. Arianfar, P. Nikander, and J. Ott. On content-centric router design and implications. In *Proc. ACM ReARCH 2010*, pages 5:1–5:6, 2010.
[6] I. Baev, R. Rajaraman, and C. Swamy. Approximation algorithms for data placement problems. *SIAM J. Comput.*, 38(4):1411–1429, Aug. 2008.
[7] W. Bellante, R. Vilardi, D. Rossi, et al. On Netflix catalog dynamics and caching performance. *IEEE CAMAD*, 2013.
[8] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *Proc. IEEE INFOCOM 2010*, pages 1478–1486, Mar. 2010.
[9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proc. IEEE INFOCOM 1999*, volume 1, pages 126–134, 1999.
[10] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino. Modeling data transfer in Content-Centric Networking. In *Proc. the 23rd International Teletraffic Congress*, pages 111–118, 2011.
[11] G. Carofiglio, V. Gehlen, and D. Perino. Experimental evaluation of memory management in Content-Centric Networking. In *Proc. IEEE ICC 2011*, pages 1–6, 2011.
[12] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker. Less pain, most of the gain: Incrementally deployable ICN. In *Proc. ACM SIGCOMM 2013*, pages 147–158, 2013.
[13] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proc. ACM CoNEXT 2009*, pages 1–12, Dec. 2009.
[14] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi. Orchestrating massively distributed CDNs. In *Proc. ACM CoNEXT 2012*, pages 133–144, 2012.
[15] J. Kangasharju, J. Roberts, and K. W. Ross. Object replication strategies in content distribution networks. *Computer Communications*, 25(4):376–383, Mar. 2002.
[16] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proc. ACM SIGCOMM 2007*, pages 181–192, 2007.
[17] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohraby. On the optimal placement of web proxies in the Internet. In *Proc. IEEE INFOCOM 1999*, volume 3, pages 1282–1290, 1999.
[18] R. Mok, E. Chan, and R. Chang. Measuring the quality of experience of http video streaming. In *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 485–492, 2011.
[19] T.-M. Pham, S. Fdida, and P. Antoniadis. Pricing in Information-Centric Network interconnection. In *Proc. IFIP NETWORKING 2013*, pages 1–9, May 2013.
[20] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou. Modelling and evaluation of CCN-caching trees. In *Proc. IFIP NETWORKING 2011*, pages 78–91. Springer-Verlag, 2011.
[21] L. Qiu, V. N. Padmanabhan, and G. M. Voelker. On the placement of web server replicas. In *Proc. IEEE INFOCOM 2001*, volume 3, pages 1587–1596, 2001.
[22] E. Rosensweig, J. Kurose, and D. Towsley. Approximate models for general cache networks. In *Proc. IEEE INFOCOM 2010*, pages 1–9, Mar. 2010.
[23] D. Rossi and G. Rossini. On sizing ccn content stores by exploiting topological information. In *Proc. 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 280–285, Mar. 2012.
[24] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie. Optimal cache allocation for content-centric networking. In *Proc. ICNP 2013*, pages 1–10, Oct. 2013.
[25] M. Yu, W. Jiang, H. Li, and I. Stoica. Tradeoffs in CDN designs for throughput oriented traffic. In *Proc. ACM CoNEXT 2012*, pages 145–156, 2012.
[26] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.