Network-friendly box-powered video delivery system

Yiping Chen, Bing Han, Jimmy Leblet, Gwendal Simon, Gilles Straub

Network-Friendly Box-Powered Video Delivery System

Yiping Chen*‡, Bing Han†‡, Jimmy Leblet‡, Gwendal Simon‡, Gilles Straub*
*Thomson Research, France
‡Institut Télécom / Télécom Bretagne, France
†Beijing University of Posts and Telecommunications, China

***Abstract*—On-demand video delivery with a Content Delivery Network (CDN) solely based on set-top-boxes has been introduced recently. This architecture combines the load balancing and fault tolerating features of peer-to-peer systems with stableness of a server based CDN by storing contents in set-top-boxes. Since the set-top-boxes usually have much longer online time than traditional PC based peers, they are ideal for On-demand video services. Currently, videos are separated into pieces and randomly stored in boxes. Requests from clients are redirected to the nearest boxes. However, random strategy may lead to some costly and inefficient allocations, for example, a video part could be placed much further than other parts of the same video, while two close boxes hosting the same part. This paper aims to reduce downloading cost by exploiting the network location of boxes when allocating the videos. We show that optimizing the allocation is NP-hard. Two locality-based online heuristics which support transient boxes are proposed. Simulations with realistic network settings demonstrate that our heuristics have comparable performance to an existing approximate algorithm and outperform the random allocation.**

I. Introduction

Most analysts forecast a continuous growth in the popularity of video on demand services and user-generated video websites [1, 2]. Along with the increasing quality of video content, these *video providers* are expected to deliver in the very next years several terabits per second of video data over the unicast IP network. Numerous papers have shown that current system architectures, based on data-centers assisted by Content Delivery Networks (CDN), are insufficient [3].

Several works have suggested to also make use of the set-top boxes or residential gateways of clients, thereafter called *boxes*, to assist the video provider [4]–[7]. The design principles of such a system have been clearly stated:

1) *a catalog of videos (noted $\mathcal{C}$) to be hosted by the box-powered CDN is selected*. Requests for the top 5% popular videos (approximately half of the global traffic) are handled by CDN proxies. The traffic generated by the three quarter most unpopular videos (a tenth of the global traffic) is managed by the data-center. Therefore, the catalog $\mathcal{C}$ is chosen in the remaining: several millions of mid-popular videos in the popularity range $[5\%, 25\%]$.
2) *a set of boxes (noted $\mathcal{V}$) agree to participate*.
3) *each video $a \in \mathcal{C}$ is cut into k_a independent substreams*. The principle of spreading distinct video substreams in distributed video service is now widely admitted [8]. The optimal number of substreams per video has been extensively studied in [6].
4) *a subset of boxes $V_a \in \mathcal{V}$ is chosen to host a substream of the video $a \in \mathcal{C}$*. The ideal number of boxes ($|V_a|$) has been thoroughly analyzed in [5, 6]. A random choice of boxes is usually assumed.
5) *the k_a substreams of a video $a \in \mathcal{C}$ is allocated to the boxes V_a*. Capacity constraint imposes to allocate only one substream to each box. Previous works have neglected the question of which substream to which box. This is the topic of this paper.
6) *requests from clients of any video $a \in \mathcal{C}$ (noted J_a) are automatically routed to the k_a nearest boxes in V_a hosting separate substreams*. Various works have addressed the nearest server selection problem [9]–[12]

The allocation of substreams to the selected boxes is crucial for network operators and Internet Service Providers (ISP). A good allocation can indeed represent a substantial savings when considering the *network cost* of the streaming sessions from their clients to the different requested substreams, *i.e.* the cost to transport the data from the boxes hosting the substreams to the client. The notion of network cost is generic: it can be computed from the inter-AS peering, the estimated traffic cost or the number of traversed routers. In this paper, without loss of generality, we use the latter indicator, so we consider Internet as a network of routers, the distance between two network elements being computed as the number of routers on the shortest path between them.

Ideally, all substreams are hosted by boxes that are in the same local network as the requester or in close sub-networks. Unfortunately, less advantageous cases may occur, where at least one substream is located much further than any other substreams. It may be quite frustrating if in the meantime two close boxes host the same substream.

In this paper, we provide a comprehensive study of the substream allocation problem. Our analysis completes the design of a massive video service with an objective that has not been yet addressed: being friendly for network operators in the sense that the whole network cost incurred by the delivery of the video content is minimized. We show in Section III that the problem of allocating optimally the substreams is NP-hard. The problem formulation and the NP-hardness is the first contribution of this paper.

Then we study two heuristics for a realistic scenario. We

consider that boxes join the system when their owners sign in. Hence, the heuristics are *online*: previous allocations can not be modified, each new box is treated iteratively. We assume that the service provider is able to estimate the location of any new box x and to determine its closest boxes. In previous works, it has been shown that the selection of $C_x \subset \mathcal{C}$ (the set of videos that the box x will be partially in charge of) can be easily decided. Then our problem is to choose, for every video a in C_x, one substream $i, i \in [1, k_a]$. We show by simulations in Section V that our heuristics have the potential to let network operators make substantial savings, which is the second contribution of the paper.

II. Related Works

Besides aforementioned works, the design of a CDN hosting multiple substreams has been introduced in [13] but this study considers only two substreams per video, so the distribution of substreams across proxies is not really an issue, and a random allocation is assumed in this work. Yet, authors provide a short description of this *coloring* problem where the goal is to assign to each proxy a color corresponding to a unique substream. More recent works have shown that increasing the number of substreams can lead to substantial performance improvements (*e.g.* [14, 15]). Therefore, the problem of allocating substreams to elements of a CDN has been more deeply analyzed, but these works usually focus on maintaining a number of substreams proportional to their popularity as a countermeasure to crashes [16, 17]. How to reduce the network cost when clients retrieve all substreams of requested videos is less frequently addressed. The distributed storage system described in [18] proposes a mechanism to discover the "closest" video substreams. However, the allocation algorithm does not take into account any network proximity, so a video substream may be excessively far from other substreams while two close boxes could store the same substream. In our context, the video service provider is expected to implement a redirection mechanism able to route the request of clients to the k *closest* boxes hosting the k required substreams of the requested video.

The deployment of CDN has often been related with a *minimum κ-Median Problem* where the problem consists of determining κ servers to host same objects so that the cost of retrieving one replica from many clients is minimized [19]. Our problem is different as now each server can choose any of the k distinct substreams, the cost of retrieving the k substreams from many clients being minimized.

III. Problem Statement and Model

In a very first attempt to formally state the problem of allocating the substreams to the boxes, we assume that the service provider is able to accurately estimate the cost $d(x, p)$ between any pair of client x and box p. It means that the service provider is able to locate the future clients of a video, which is actually a very strong assumption. The decision problem related with allocating the substreams to boxes can be formulated as follows. As we consider separately each video, we will omit in the following the subscript a in the notation. The set of clients is thus noted J, the set of selected boxes V and the number of descriptions k.

Decision Problem for Box-powered Video Delivery Network

INSTANCE : Two subsets J and V, a positive integer $k \leq |V|$, a cost function $d : J \times V \to \mathbb{R}^+$ and a positive real $K \in \mathbb{R}$.
QUESTION : Is there a labelling function φ from V onto $\{1, \cdots, k\}$ such that the sum of all minimum costs to other substreams is lower or equal to K, that is :

$$\sum_{x \in J} \sum_{i \in [1,k]} \min \{d(x, p) : p \in V, \varphi(p) = i\} \leq K$$

This problem is a subproblem of k-Product Uncapacitated Facility Location Problem (k-PUFLP) [20]. It is known as NP-hard. An approximate optimal algorithm for the k-PUFLP problem has been recently proposed in [21]. The algorithm is able to find a solution which it is at most $\frac{3}{2}k - 1$ times the optimal solution.

Unfortunately, the client location can hardly be predicted, moreover this algorithm requires to recompute the whole allocation after every event, for example a client appears or a box crashes. Therefore it can not be realistically implemented. It can however be useful as a competitor for any algorithms fitting better with realistic settings. Indeed, in a snapshot of our simulations, it is possible to compare the allocation given by any heuristic versus this proven approximation of the optimal solution.

A. Our Approach

Actually the only parameters that we can assume to be known by the video provider are (i) the local network of the box, *i.e.* the identifier of the first router that connects this box to the Internet, and (ii) the network cost between every two routers (see [22]–[24] for some techniques that may be used). Our claim is that, when the network cost between two boxes is low, there is a high probability that both boxes will be chosen simultaneously among the k nearest boxes of a given client. This claim could be formulated as follows. We define the matrix of *remoteness* $\mathcal{D} = (\delta_{p,q})_{p,q \in V}$ to represent the closeness of two boxes. In a first implementation, we choose $\delta_{p,q}$ as a linear function of the cost $d(p, q)$. The smaller is $\delta_{p,q}$, the higher is the probability that they will be chosen simultaneously by a box, hence the more likely they should host distinct substreams.

We validate this claim through a short simulation. We give more details about the parameters of the simulation environment in Section V, that is, we just give here some basic elements. We are using a router map, namely *nec* topology, and we attach some boxes and some clients to some end-routers in a realistic manner. Then we determine, for each client, the k nearest boxes. For each pair of distinct boxes p and q, we compute three values: a_{pq}, the number of clients having either p or q among their k nearest boxes; b_{pq}, the number of clients having both p and q in their k nearest boxes; and d_{pq}, the hop distance between p and q. Then, for each distance value

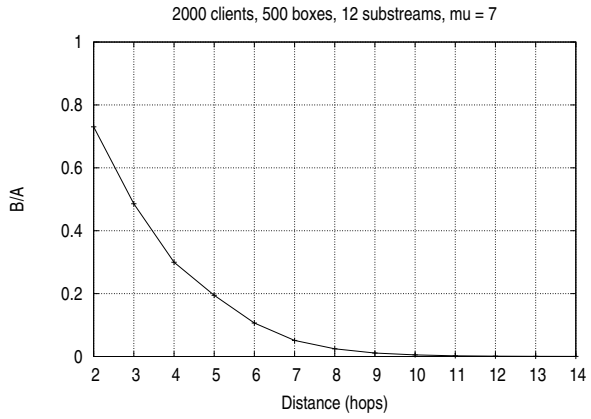

Fig. 1. Correlation between the distance between two boxes and the probability that they are both among the k nearest of a client

d (from 1 to the network diameter), we compute $\frac{B(d)}{A(d)}$, where $A(d) = \sum_{(p,q):d_{pq}=d} a_{pq}$, $B(d) = \sum_{(p,q):d_{pq}=d} b_{pq}$. A point at $(5, 0.2)$ means that 20% of boxes at distance equal to 5 are chosen simultaneously by a client.

Results presented in Figure 1 show a strong correlation between the distance between boxes and the probability that they are selected simultaneously by the same client. When $d = 2$, the edge routers of two boxes are connected to the same sub-network. As can be expected, this case exhibits the more significant probability while the probability is strictly decreasing. Please observe also that this decreasing is quite quick, even boxes that are not so far (*e.g.* seven routers far) have very few chances to be selected together by a client. Note also that two close boxes are not sure to be selected together because, for some clients, this pair of boxes could be ranked at k and $k + 1$.

Thus, our approach is intuitive: we try to allocate distinct substreams to close boxes, of course. But, more important, we try to allocate the subsreams such that every box is close to *all* other substreams, that is, a client has higher probability to find all substreams among its closest boxes.

B. Rainbow Cost Problem Formulation

Now, we define the *rainbow cost* for a box $p \in V$ as the sum of costs between p and the k boxes assigned to the k substreams with the lowest remoteness. A box p chooses itself for its substreams because $\delta_{p,q} = 0$. Formally, the rainbow cost of p is:

$$\sum_{i\in[1,k]} \min\{\delta_{p,q} : q \in V, \varphi(q) = i\}$$

Ideally, we would like to push k substreams of the same video such that each box and its $k-1$ boxes having the lowest remoteness collectively store all k substreams. However this is not always possible and under this case, we would naturally try to minimize the sum of rainbow costs. The decision problem associated with this goal is as follows.

Minimal Overall Rainbow Cost Problem

INSTANCE : A set V, a positive integer $k \leq |V|$, a symmetric function $\mathcal{D} : V \times V \to \mathbb{R}^+$ and a positive real $K \in \mathbb{R}$.

QUESTION : Is there a labelling function φ from V onto $\{1, \cdots, k\}$ such that the sum of all rainbow costs is lower or equal to K, that is

$$\sum_{p\in V} \sum_{i\in[1,k]} \min\{\mathcal{D}(p, q) : q \in V, \varphi(q) = i\} \leq K$$

By solving this rainbow cost problem, we come up with a substream distribution scheme where each box is connected with $k - 1$ boxes hosting complementary substreams and the sum of the costs on the edges are minimized. This result can be easily extended to the original problem (with clients) in the following sense. Actually, an edge between two boxes reflects the likeliness that both of them are close to the same client. Thus if a client is close to a given box p, it is likely to be close to the boxes that are close to p. If all substreams are stored in these boxes, the allocation has great chance to be near an optimal solution. Sub-optimal results are possible when it is impossible to allocate all substreams on the nearest neighbors of every client. However, when this occurs, it is often possible to allocate these substreams on a slightly extended set of nearer neighbors.

C. NP-Completeness of Rainbow Cost Problem

The Rainbow Cost Problem is also a subproblem of the k-PUFLP problem, which is NP-Complete. But as it is a strict subproblem (we can not transform any k-PUFPL instance into an instance of Rainbow Cost Problem), we must prove that it is also NP-Complete. This problem is tight with the *Domatic Partition Problem* [25] which is among the classical NP-Complete problems. A *dominating set* in a graph is a set of vertices such that every vertex of the graph is either in this set or has a neighbor in this set. A *domatic partition* is a partition of vertices such that each set of the partition is a dominating set. Thus, the domatic partition problem is to determine if a graph admits a domatic partition of size K, where the size of a domatic partition is the number of sets of the partition.

Domatic Partition Problem

INSTANCE : A graph $G = (V, E)$ with V the set of vertices and E the set of edges, and a positive integer K.

QUESTION : Is there a partition D of V with K sets such that each set belonging to D is a dominating set of G ?

Theorem 1 *Rainbow Cost Problem is NP-Complete.*

Proof. Given an instance of Rainbow Cost Problem and a labelling function φ, verifying that this is a valid one is clearly polynomial in the size of the problem: hence the Rainbow Cost Problem belongs to NP.

We now reduce Domatic Partition Problem to our problem. Given a graph $G' = (V', E')$ and a positive integer K' as an instance of the Domatic Partition Problem, we define the

instance of Rainbow Cost Problem as follows : V as V', k as K', K as $K' \cdot |V'|$ and for any $p, q \in V$, we define $\mathcal{D}(p,q) = 1$ if p is a neighbor of q in G', $\mathcal{D}(p,q) = 1$ if $p = q$ and $\mathcal{D}(p,q) = K' \cdot |V'| + 1$ otherwise. Clearly the instance of the Rainbow Cost Problem can be constructed in polynomial time in the size of the instance of Domatic Partition Problem. We claim that G' admits a domatic partition of size K' if and only if our instance admits a labelling φ from V to $\{1, \cdots, k\}$ such that the sum of all rainbow costs is lower or equal to K.

For the forward implication, assume that D' is a domatic partition of G' having K' elements. Let $D' = \{D'_1, \cdots, D'_{K'}\}$, then for any $p \in V$ we define $\varphi(p)$ such that $p \in D'_{\varphi(p)}$. As $k = K'$ and as D' is a partition of V, we obtain that φ is a labelling φ from V to $\{1, \cdots, k\}$. Now, as every element of D' is a dominating set, for any $p \in V$ and for any $s \in \{1, \cdots, k\}$, we obtain that $\min\{\mathcal{D}(p,q) : q \in V, \varphi(q) = s\} = 1$. So we obtain that for every element p of V its rainbow cost is exactly k. Thus φ provides a labelling such that the sum of all rainbow costs is $k \cdot |V'|$ which is lower or equal to K.

For the backward implication, assume that φ is a labelling function such that the sum of all rainbow costs is lower or equal to K. Let D'_c the set $\{p \in V : \varphi(p) = c\}$. We claim that $D' = \{D'_c : c \in \{1, \cdots, k\}\}$ is a domatic parition of G' having K' elements. First of all, as φ is a function from V onto $\{1, \cdots, k\}$, we obtain that each D'_c is not empty and thus that D' is partition of V'. Second, we show that each D'_c is a dominating set of G'. Assume on the contrary, that there exists a vertex $p \in V'$ such that p is neither in D'_c nor has a neighbor in D'_c. Thus, by definition of $\mathcal{D}$, we obtain that $\min\{\mathcal{D}(p,q) : q \in V, \varphi(q) = c\} = K' \cdot |V'| + 1$, and then the rainbow cost of p is greater than $K' \cdot |V'| + 1$. This contradicts that φ is a labelling such that the sum of all rainbow costs is lower or equal to $K = K' \cdot |V'|$. □

IV. Proposed Heuristics

The approximate algorithm for the k-PUFLP gives guaranteed results, but can not be realistically implemented. The simplified problem for realistic implementations is unfortunately NP-hard. We define in the following two *online* heuristics that have no guaranteed approximation results but are expected to produce good results and have practical interest in large-scale dynamic systems.

Let say that box p' is the *nearest* to box p if $\forall q \in V, \delta_{pp'} \leq \delta_{pq}$. The set of boxes $K(p)$ contains the k nearest boxes to p. For our problem, the optimal overlay would be the merge of the following two graphs:

- a k-nearest neighbor graph, *i.e.* two boxes p, q are linked if either $q \in K(p)$, or $p \in K(q)$
- a *rainbow colored* graph, *i.e.* each box and the set of its neighbors are assigned with all colors in C.

Unfortunately, the k-nearest graph is not necessarily a *domatically full* graph [25], in other words the existence of a partition into $k+1$ dominating sets is not guaranteed. This implies that a rainbow coloring with $k+1$ colors on the k-nearest graph is not guaranteed. Therefore, the idea of the heuristics is either to relax the distance to the furthest boxes a box can reach in a rainbow colored graph, or to allow sub-optimal rainbow coloration of a nearest neighbor graph.

Assume a box p is able to sort all boxes in ascending distance to itself. We note $r_p(q)$ the *rank* of the box q for p, meaning that a number of $r_p(q)$ boxes are closer to p than q. We denote by $r(p)$ the rank of the nearest box to p completing, with all closer boxes, the set of colors. Formally, $r(p)$ is the smallest integer ρ such that $\{\varphi(q) : q \in V, r_p(q) \leq \rho\} = C$. We note r the maximal value of $r(p)$ for all boxes p in V. The reduction of the value of r is an objective. As previously said, an optimal overlay would be obtained with $r = k$.

There is another motivation for reducing r. Intuitively, the smaller is r, the more "local" is the required knowledge to build the overlay. We observe indeed that most services aiming at discovering nearby entities (*e.g.* [22]) succeed in quickly determining a kind of "position" and the entities that are also near this position. But the discovery of further entities is both harder and less accurate. Even for a central server monitoring the system, exploring distant boxes, for instance boxes in other AS, requires computation time and is not likely to provide relevant results.

A. Rainbow Coloring Heuristic

The *Rainbow Coloring Heuristic* builds an overlay where a box and its nearest neighbors contain all colors. The idea we suggest is simple. When a new box joins the system, it iteratively explores the nearest boxes until it discovers $k-1$ colors, then it chooses the unassigned color.

This heuristic is definitely more efficient in the case of a metric, the triangular inequality enlarging the probability that some close neighbors will also be satisfied by this new color. We note however that this heuristic does not provide any bound on r. In the worst case, a new box has to explore all other boxes in order to determine its own color.

This heuristic corresponds to our first approach: we relax the distance to the furthest box completing the rainbow coloring.

B. k-Nearest Partially Colored Heuristic

The *k-nearest Partially Colored Heuristic* builds a k-nearest neighbor overlay, and tries to rainbow color it. A new box p first looks for its k closest boxes $K(p)$, and selects one color among all colors that are not assigned in $K(p)$. There is ideally only one free color, but, as previously said, the k-nearest neighbor graph may not be a domatically full graph, so some boxes should probably choose, by default randomly, among more than one color.

There is still no bound on r in this heuristic because there is no guarantee on the rainbow coloration of the neighborhood. But the computation of this heuristic ensures that a box should not determine more than its k nearest boxes, so the computation cost is bounded to the discovery of k boxes, while the rainbow coloring heuristic has a computation cost that is bounded by the number of boxes.

Note that this k-nearest partially colored heuristic as well as the random allocation may generate overlay of boxes where no

Dataset	Year	Type	# Nodes	# Links
GT-ITM	–	Generated	1944	6584
HOT	–	Generated	939	998
Nec	2003	Real	47k	119k
CAIDA	2003	Real	192k	600k

TABLE I
TOPOLOGY DATASETS

box at all chooses a certain color. In the evaluation part, we tackle this issue by first allocating all colors to k randomly chosen boxes, then assigning to other boxes through this heuristic.

C. Discussion

Various more complex and efficient heuristics could be designed, but we are looking for simple heuristics with minimal requirements. We note here that the k-nearest Partially Colored heuristic does only require the knowledge of the k nearest boxes, which can be easily done in a centralized service or using a dedicated network discovery service. The Rainbow Coloring heuristic is more costly as more than k boxes have to be discovered with no prior guarantee on the exact number of required neighbors. Evaluations will however reveal that this number of boxes is quite close to k in the real network, so the cost of this heuristic is generally acceptable.

From a theoretical point of view, the challenge is to build a rainbow colored overlay where the furthest neighbor of a box is as close as possible. As previously shown, this problem corresponds to finding the minimum integer κ, independent of the number of nodes, such that the κ-nearest neighbor graph is guaranteed to have a domatic number greater than k. To the best of our knowledge, theoretical studies related with domatic number have explored neither k-nearest neighbor graphs, nor more general proximity or spanner graphs.

V. PERFORMANCE EVALUATION

We conduct extensive simulations to validate the efficiency of the proposed heuristics. The simulations rely on realistic network topologies, widely adopted parameters and experimentally demonstrated properties, in order to obtain accurate and meaningful results.

A. Simulation Environment

Our simulation platform is PeerSim, which emulates a vast network expected to reflect the main characteristics of the Internet. A router-level topology resulted from the *Network Cartographer (nec)* [26] is used as our default network topology, but three other network topologies have been also used in the simulator: (i) *GT-ITM* [27], which is not recent though still often used. It incorporates hierarchical structures observed in Internet, (ii) *HOT (Heuristically Optimal Topology)* [28], which includes technology constraints and economic considerations, and (iii) *CAIDA* [29], which is considered as the reference Internet map for researchers. The main characteristics of these topologies are listed in Table I.

In our context, all system elements (clients and boxes) are located at the edge of the Internet. Since nodes in the topology graph represent routers, we randomly select a number of edge routers with degree equal to one. Then, we transform them into a "virtual" small cluster according to [30]: we attach to a selected edge router a number of elements following a normal distribution $N(\mu, 0.2\mu)$ where μ is an adjustable parameter. Intuitively, a high value for μ means that elements are clustered, which is representative of a VoD service that either is offered to clients of a few ISP, or is a local service, for instance delivering video related with local topics. On the contrary, a low value for μ indicates that elements are spread all over the Internet, so the VoD service is probably a worldwide service. Among these elements, some are randomly selected to be boxes.

It is now time to set some default parameters for the ratio of boxes, the number of clients and the parameter μ. We rely on some trustworthy trends highlighted by recent measurement studies [31]. We envision a service which is accessed one billion time a day, so approximately three million typical four-minutes long videos are simultaneously downloaded. The size of the catalog of mid-popular videos can be approximated with twenty million of videos, these videos being watched simultaneously by one million users. The number of boxes agreeing to participate as well as the amount of resources (storage and bandwidth) they agree to supply are difficult to estimate. But, in average for one video, with the same context than described in [6], we consider by default that 30 boxes host the video, and that this video is accessed by $1,500$ clients. This latter number does not mean simultaneous clients, but rather over the time this video is stored on the box until it is replaced by another video. This represents around hundred giga-bytes of data and less than one megabits of upload bandwidth. Finally, the number of clients for one video in this catalog is low in comparison with the total number of clients that use the service. We use equivalent ratio to set $\mu = 7$, that is, seven elements are, in average, active (either as client, or as box) about one video in this catalog of mid-popular videos in a local network.

The optimal number of substreams in distributed environment is still an open question since the seminal works [14], *e.g.* k is fixed at 15 in [5] or at 8 in [18]. However, when it applies specifically for our context of box-powered CDN, it appears that the number should preferentially be chosen in the range $k = 8 \ldots 15$ [6]. In our simulations, we use the default value of $k = 12$, but k varies in a range from 1 to 30.

Given a substream allocation method, every client should simultaneously download k distinct substreams. The problem of selecting the nearest servers over Internet has been addressed in many previous works [9]–[12]. The commonly used selection metrics are round-trip time (RTT), hop count and available bandwidth. Hop count is used in [11, 12] rather than RTT, because backbone links are often faster than local links, the use of RTT could favor traversing backbone links. Studies on *peer selection* (distinct from *server selection*) have also adopted hop count as metric [32], because it reflects

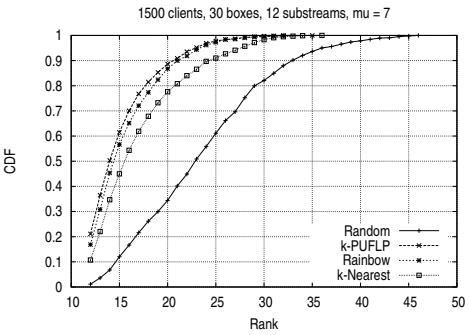

Fig. 2. Cumulative Distribution Function of the rank of the furthest box a client should explore before to reach all substreams

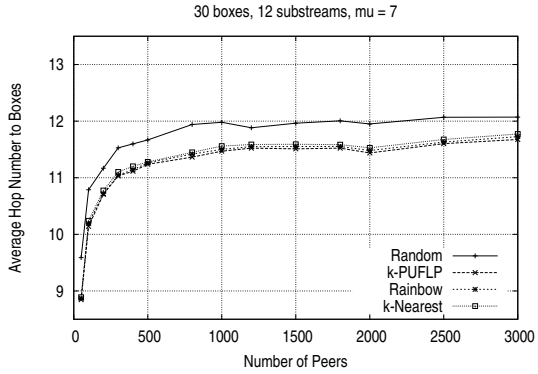

Fig. 3. Average distance (nb. of routers) when the number of clients increases

directly network proximity and network resource utilization. Additional metrics could be combined with hop count to achieve substream selection, such as RTT, loss rate and server load. Here, we make trade off between simulation complexity and reality by using hop count based selection method, as used also in [13]. Thus, every client picks the k closest distinct substreams, the distance to substream being the hop-number in the topology graph.

B. Heuristic Performance

We evaluate the efficiency of the heuristic by observing the *rank* of the furthest box a client should visit to complete its k substreams. That is, a perfect assignation would produce that the k nearest boxes of every client contain the k distinct substreams. On the contrary, when the j^{th} nearest box to a client stores the same substream as the i^{th} one, with $j > i$, the allocation is not optimal. That is, we measure the rank of the furthest box that is required to explore before to reach all substreams. Figure 2 demonstrates the Cumulative Distribution Function (CDF) of clients versus the rank of their furthest box. A point at $(20, 0.9)$ in the figure could be understood as 90% of the clients have found the k substreams among their 20 nearest boxes. Rainbow, k-Nearest, k-PUFLP algorithms are compared with the Random allocation. As expected, the k-PUFLP heuristic achieves the best placement efficiency, while the random allocation exhibits the worst results. In the case of 12 substreams, more than half of clients have to connect to at least their 23^{rd} nearest boxes to reach 12 substreams, which means more than half of the explored boxes are redundant. In this simulation, both rainbow and k-Nearest heuristics have similar results compared to k-PUFLP. Rainbow coloring heuristic is slightly better than k-Nearest.

Besides measuring the efficiency of the heuristics, note that exploring more boxes to reach all substreams can also be costly for the clients when connecting to the system. Therefore, reducing the number of boxes to discover before to start downloading the content can enhance the start-up delay and reduce the cost of running a discovery service.

C. Network Cost Reduction

The network cost is defined as *average distance client to boxes* and is obtained as follows: we measure the overall number of routers that are in the paths between each client and its k closest boxes having the k substreams, then we divide this value by the number of boxes.

The impact of various system parameters on the network cost is studied by varying the number of clients, the number of boxes, the number of substreams and the density of elements μ in each simulation run. The results are shown in Figure 3 to Figure 6, respectively.

We immediately observe that both heuristics always outperform the random allocation and comparable to k-PUFLP algorithm under various settings. It proves not only the assumption that two close boxes are likely to serve the same clients, so should be complementary, but also the successful rainbow coloring of the overlay. However, if we look further at the results, we find that the difference between the random allocation and the other heuristic is not so significant. That is, the gain is usually around less than a half router over an average of 11 routers. This difference can become substantial in the case of large-scale systems, but it is however quite modest. Therefore, although our results are convincing about the performance of our heuristics and the general approach detailed in this study, the interest of implementing such techniques in a real implementation can be considered. We now let focus on the behavior of the system.

In Fig. 3, the average distance is quite stable while the number of clients increases in the system. This curve reveals the scalability of the system in the sense that the traffic stays local: boxes find their substreams in their surroundings.

In Fig. 4, the average distance decreases when the number of boxes increases. All heuristics have obviously the same result for 12 boxes because we have 12 substreams so each box has a different substream. We observe that the gain of our heuristics to the random allocation reaches a kind of optimality for 40 boxes. This has also been confirmed by other simulations. In

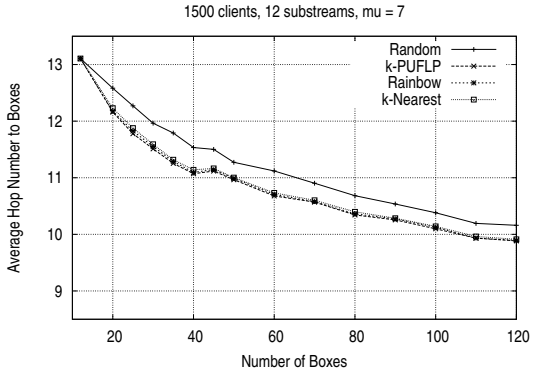

Fig. 4. Average distance (nb. of routers) when the number of boxes increases

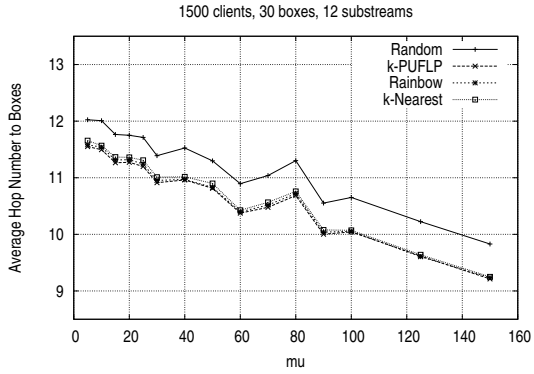

Fig. 6. Average distance (nb. of routers) when the density of boxes increases on the edge routers

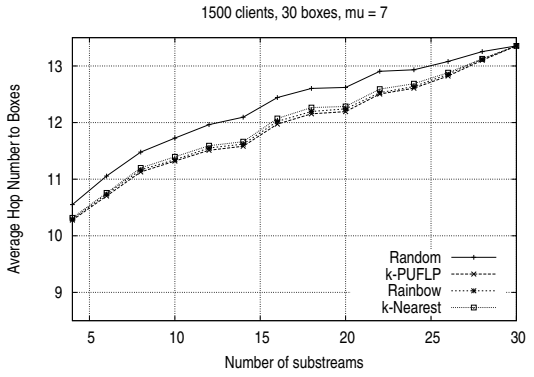

Fig. 5. Average distance (nb. of routers) when the number of substreams increases

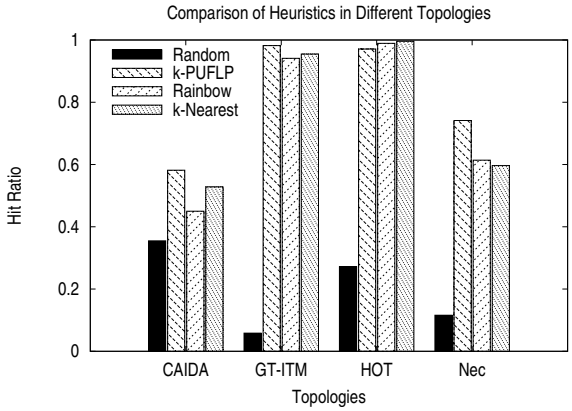

Fig. 7. Probability to reach k substreams among the k closest boxes in various network topologies

general, maximal gain is achieved when the number of boxes is approximately equal to three times the number of substreams. It can be explained by noting that, when the number of boxes becomes high, a client can find many boxes at a reasonable distance to it. Whatever the cleverness of the heuristic, it is likely that this large set of boxes will be assigned to all colors. Therefore, the more boxes there are, the less advantageous will be the heuristics because all colors will be easily found nearby when one assigns the colors randomly.

In Fig. 5, the network cost increases with the number of substreams, which is quite natural because, with a fixed number of boxes, the clients should necessarily reach far boxes to complete their substreams. We observe again the optimal number of boxes per substream is about 3. Actually, when the number of substreams is equal to the number of boxes, all heuristics have obviously the same results. The other extreme scenario occurs when the number of substreams is equal to one, where all heuristics will again obviously give the same results. Between these two extrema, the heuristics exhibit a gain that seems to reach its maximal value when the number of boxes is between $k * 2$ and $k * 3$. It also corresponds to a network cost which appears to be a good trade-off in both Fig. 4 and 5, where we observe that the derivative of the curve is maximal for a number of boxes between 25 and 40 in Fig.4 and a number of colors ranging from 10 to 17 in Fig.5.

Finally, in Fig. 6, the clustering density of boxes evolves. In a highly clustered service, which may emulate a local service, the average distance to box is naturally decreasing as some boxes are probably attached to the same edge router as the clients, so the distance is null to reach this substreams. Substantial gain can be obtained by a clever algorithm which can avoid that two boxes attached to the same routers store the same substream, so the gain tends to slowly increase.

D. Impact of Network Topology

We now aim to show that these heuristics are not opportunistically presented in these simulations because they fit with the selected network topology. We emphasize in Fig. 7 that these heuristics can provide a substantial gain even when the distance is computed by the number of traversed AS or in a topology exhibiting various characteristics. We have run

the algorithms on top of topologies described in Tab.I. The metric called *hit ratio* is the probability that a client finds the k substreams among its k nearest boxes. The results confirm the previous curves by showing that our heuristics always outperform the random allocation, with results that are almost similar for the Rainbow heuristic and k-nearest partially colored heuristic and both are very close to the unrealistic k-PUFLP algorithm. As the k-nearest Partially Colored Heuristic is less costly to implement than the Rainbow Colored one, it is definitely a good candidate to replace the random allocation.

VI. Conclusion and Future Works

Long videos encoded into multiple substreams and box-powered CDN are two major trends for the future of video services. An important but less studied problem is how to store these substreams in proxies such that the overall download cost is minimized when the client has to retrieve a set of substreams. We have identified and formulated this problem and proven its NP-completeness. Two fast and distributed heuristics have been proposed. The heuristics are based on the observation that if a client selects a certain proxy for one substream, it is very likely that it selects the neighboring proxies of the already selected one. The proposed heuristics have been compared with the existing k-PUFLP and Random placement algorithms. On one hand, both proposed heuristics demonstrated promising performance gains over the random placement, while on the other hand, they have comparable performance as the k-PUFLP, which has bounded performance guarantee but requires perfect knowledge about the network. In contrast, our heuristics only require knowledge about the proxies which is more realistic. The simulations also demonstrate that an optimal gain over Random placement is achieved when the number of proxies is about 3 times the number of substreams which may shed light on the way to better algorithms.

References

[1] A. Amel, "The user-generated online video market," Screen Digest, Tech. Rep., 2008.

[2] M. Inouye, "User Generated Content–More than Just Watching the YouTube and Hangin' in MySpace," In-Stats Consumer Media and Content Service, Tech. Rep., 2007.

[3] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 133–144, 2007.

[4] M. Allen, B. Zhao, and R. Wolski, "Deploying Video-on-Demand Services on Cable Networks," in *IEEE Int. Conf. on Distributed Computing Systems (ICDCS)*, 2007.

[5] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot, "Achievable catalog size in peer-to-peer video-on-demand systems," in *Int. Workshop on Peer-To-Peer Systems (IPTPS)*, 2008.

[6] A. Nafaa, S. Murphy, and L. Murphy, "Analysis of a Large-Scale VOD Architecture for Broadband Operators: A P2P-Based Solution," *IEEE Communications Magazine*, December 2008.

[7] K. Suh, C. Diot, J. Kurose, L. Massoulié, C. Neumann, D. F. Towsley, and M. Varvello, "Push-to-peer video-on-demand system: Design and evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1706–1716, 2007.

[8] O. Campana, R. Contiero, and G. Mian, "An H. 264/AVC Video Coder Based on a Multiple Description Scalar Quantizer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 2, pp. 268–272, 2008.

[9] S. Dykes, K. Robbins, and C. Jeffery, "An empirical evaluation of client-side server selection algorithms," in *IEEE INFOCOM*, 2000.

[10] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *IEEE INFOCOM*, 2002.

[11] J. D. Guyton and M. F. Schwartz, "Locating nearby copies of replicated internet servers," *SIGCOMM Comput. Commun. Rev.*, vol. 25, no. 4, pp. 288–298, 1995.

[12] J. Almeida, D. Eager, M. Vernon, and S. Wright, "Minimizing delivery cost in scalable streaming content distribution systems," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 356–365, 2004.

[13] J. Apostolopoulos, T. Wong, W. tian Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *IEEE INFOCOM*, 2002.

[14] Y. Shen, Z. Liu, S. Panwar, K. Ross, and Y. Wang, "Streaming layered encoded video using peers," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2005.

[15] X. Xu, Y. Wang, S. Panwar, and K. Ross, "A peer-to-peer video-on-demand system using multiple description coding and server diversity," in *Int. Conf. on Image Processing (ICIP)*, 2004.

[16] T. T. Do, K. A. Hua, and M. A. Tantaoui, "Robust video-on-demand streaming in peer-to-peer environments," *Comput. Commun.*, vol. 31, no. 3, pp. 506–519, 2008.

[17] L. Guo, S. Chen, and X. Zhang, "Design and evaluation of a scalable and reliable p2p assisted proxy for on-demand streaming media delivery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 5, pp. 669–682, May 2006.

[18] W.-P. Yiu, X. Jin, and S.-H. Chan, "Vmesh: Distributed segment storage for peer-to-peer interactive video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 9, pp. 1717–1731, Dec. 2007.

[19] J. Kangasharju, K. Ross, and J. Roberts, "Performance evaluation of redirection schemes in content distribution networks," *Computer Communications*, vol. 24, no. 2, pp. 207–214, 2001.

[20] A. Klose and A. Drexl, "Facility location models for distribution system design," *European Journal of Operational Research*, vol. 162, no. 1, pp. 4–29, 2005.

[21] B. Li and R. Li, "Approximation algorithm for k-PUFLPN," *Computer Engineering and Applications (in Chinese)*, vol. 44, no. 1, pp. 97–99, 2008.

[22] B. Wong, A. Slivkins, and E. G. Sirer, "Meridian: a lightweight network location service without virtual coordinates," in *ACM SIGCOMM*, 2005.

[23] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute," in *Usenix Internet Measurement Conference (IMC)*, 2006.

[24] A. Boudani, Y. Chen, and G. Simon, "A quicker way to discover nearby peers," in *ACM CoNEXT*, 2007.

[25] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan, "Approximating the domatic number," *SIAM J. Comput.*, vol. 32, no. 1, pp. 172–195, 2002.

[26] D. Magoni and M. Hoerdt, "Internet core topology mapping and analysis," *Computer Communications*, vol. 28, no. 5, pp. 494–506, 2005.

[27] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *IEEE INFOCOM*, 1996.

[28] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the internet's router-level topology," in *ACM SIGCOMM*, 2004, pp. 3–14.

[29] [Online]. Available: www.caida.org

[30] B. Yang and H. Garcia-Molina, "Designing a super-peer network," in *Int. Conf. on Data Engineering (ICDE)*, 2003.

[31] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proc. of ACM SIGCOMM conference on Internet measurement*. ACM New York, NY, USA, 2007, pp. 1–14.

[32] M. Castro, P. Druschel, Y. Hu, and A. Rowstron, "Exploiting network proximity in peer-to-peer overlay networks," Microsoft Research, Tech. Rep. MSR-TR-2002-82, May 2002.