More Control Over Network Resources: an ISP Caching Perspective

Daphne Tuncer, Marinos Charalambides, Raul Landa, and George Pavlou
Department of Electronic and Electrical Engineering
University College London
Email: {d.tuncer, m.charalambides, r.landa, g.pavlou}@ee.ucl.ac.uk

***Abstract*—Management operations performed by Content Delivery Network (CDN) providers consist mainly in controlling the placement of contents at different storage locations and deciding where to serve client requests from. Configuration decisions are usually taken by using only limited information about the carrier networks, and this can adversely affect network usage. In this work we propose an approach by which ISPs can have more control over their resources. This involves the deployment of caching points within their network, which can allow them to implement their own content placement strategies. The work presented in this paper investigates lightweight strategies that can be used by the ISPs to manage the placement of contents in the various network caching locations according to user demand characteristics. The proposed strategies differ in terms of the volume and nature of the information required to determine the new caching configurations. We evaluate the performance of the proposed strategies, in terms of network resource utilization, based on a wide range of user demand profiles and we compare the obtained performance according to metrics we define to characterize the demand. The results demonstrate that the proposed metrics can provide useful indications regarding the performance one strategy can achieve over another and, as such, can be used by the ISP to improve the utilization of network resources.**

I. Introduction

Content Delivery Networks (CDNs) have been the prevalent method for the efficient delivery of rich content across the Internet. In order to meet the growing demand for content, CDN providers deploy massively distributed storage infrastructures that host content copies of contracting content providers and maintain business relationships with ISPs. Surrogate servers are strategically placed and connected to ISP network edges [1] so that content can be closer to clients, thus reducing both access latency and the consumption of network bandwidth for content delivery.

Current content delivery services operated by large CDN providers like Akamai [2] and Limelight [3] can exert enormous strain on ISP networks [4]. This is mainly attributed to the fact that CDN providers control both the placement of content in surrogate servers spanning different geographic locations, as well as the decision on where to serve client requests from (i.e. server selection) [5]. These decisions are taken without knowledge of the precise network topology and state in terms of traffic load and may result in network performance degradation.

In this work we propose a cache management approach with which ISPs can have more control over their network resources. Exploiting the decreasing cost of storage modules, our approach involves operating a limited capacity CDN service within ISP networks by deploying caches at the network edges (Fig.1). These can be external storage modules attached to routers or, with the advent of flash drive technology, integrated within routers. Such a service can cache popular contents, specific to an ISP, and serve most client requests from within the network instead of fetching content items from surrogate/origin servers. Empowering ISPs with caching capabilities can allow them to implement their own content placement and server selection strategies which will result in better utilization of network resources. In addition, there are economic incentives for an ISP to adopt this approach given that traffic on inter-domain links can decrease significantly. In order to deploy such an approach, new interaction models between ISP and CDN providers may need to be defined. These would address issues relating to the resolution of content requests and the exchange of necessary information, e.g. content items to cache, their popularity and size. An alternative solution for obtaining content popularity/size information would involve the ISP maintaining records of previous user requests from which it can infer future demand. This prediction could be enhanced with information from other sources like other ISPs as well as CDNI [6]. Although these are challenging issues, in this paper we focus on resource management mechanisms and we plan to investigate these issues in the future.

While the utilization of network resources is affected by both content placement and server selection operations, this work concentrates on the former. Research CDNs, such as Coral [7], have proposed distributed management approaches [8]. However, commercial CDNs have been traditionally using centralized models for managing the placement of content in distributed surrogate servers. Complex algorithms are executed in an off-line fashion for determining the optimal placement of content copies for the next configuration period (typically in the order of days). With the objective of keeping the cost and the complexity of the approach low, we employ in this study simple strategies to decide on the placement of content copies in the various caching points. Such strategies do not incur significant processing and communication overhead among distributed decision points and their functionality can thus be realized by commodity hardware components. In contrast to the ISP-centric caching approaches proposed in [9] and [10], which exclude CDNs from the delivery chain, our solution maintains the CDN presence but only for the purpose of providing content which is not locally cached by an ISP. This will ensure content

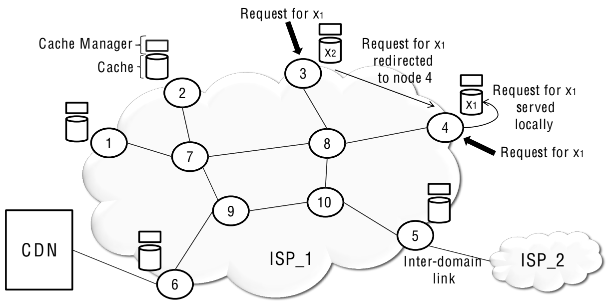

Fig. 1. Overview of the proposed caching infrastructure

availability given the global footprint of large CDN providers.

A distributed in-network cache management approach has also been proposed in [11], which focuses on content *replacement* strategies operated at short time-scales. Our approach is more geared towards longer re-configuration periods (in the order of hours) and can be complemented by online solutions that optimize the inital content placement configuration in the face of dynamic changes in user demand. We propose two categories of strategies that can be used by an ISP to control the placement of content at the different network caches. In the first category, only information about the demand from locally connected users is used to determine the configuration of each of the caches, whereas in the second knowledge about global content popularity and interests in the different caching locations is required. In order to analyze how the proposed strategies can be affected by user demand, we define specific metrics to characterize the user demand profile. We evaluate the performance of the different strategies in terms of network resource utilization according to the proposed metrics and discuss the performance obtained over a wide range of profiles.

The rest of the paper is organized as follows. Section II provides an overview of the proposed approach and introduces the main modeling assumptions of this work. Section III presents the details of two categories of content placement strategies. Section IV describes the methodology used to evaluate the performance of the proposed strategies and analyzes the evaluation results. Section V discusses related work. A summary of the work and future directions is finally presented in Section VI.

II. Content Distribution Management

A. Problem Statement

The scenario considered in this paper is depicted in Fig.1. A set of caches are deployed at the network edges, so that a cache is associated with every network edge node. In addition, each cache is locally controlled by a cache manager which is responsible for enforcing caching decisions.

Unlike the heavyweight caching infrastructure maintained by CDN providers, the proposed solution allows the ISP to operate a simpler and lightweight caching service. In this case, the total caching space available in the network may not permit in practice to store all possible contents provided by the CDN. A subset of the contents that the ISP wishes to cache in its network (for instance the most popular ones) need therefore to be pre-selected, so that the volume of selected contents is smaller or equal than the total caching capacity. In this work we assume that information about the content items (e.g. popularity, size) could be available to the ISP, which could be provided by the CDN or directly inferred by the ISP. Since the total number of contents that can be stored in each cache depends on the cache capacity, it may not be possible to accommodate all the contents at each individual caching location. In this case, the contents are distributed across the different caching locations, so that each content is stored in at least one cache in the network. As a result, user requests for content can be served from within the network instead of being redirected to CDN servers. As depicted in Fig.1, if a request for content x_1 is received at edge node 4 and content x_1 is available in the local cache, it is served locally. Otherwise, the request is redirected to one of the caches in the network where x_1 is stored. For instance, a request for content x_1 received at node 3 is redirected to and served by node 4.

Serving a content request from the same location as the request directly from the access node (i.e. locally) does not affect the utilization of network resources. Retrieving, however, a content from a remote caching location generates network traffic. Intelligently managing the placement of the contents in the different caches, under specific user demand characteristics, can therefore allow an ISP to control the utilization of network resources.

B. Intelligent Content Placement Approach

Given a set of M caches m with capacity c_m, and a list of X contents x with size s_x, controlling the placement of contents consists in determining the configuration of each of the caches, according to the current user demand, so that the network resources can be used more effectively. More specifically, the placement problem is to determine the number of copies of each content to store in the network, as well as the location of each copy, so that the two following constraints are satisfied: 1) the placement of contents in each individual cache satisfies the cache capacity constraint, and, 2) each content is cached in at least one caching location. The resulting content placement is then used as an input by a server selection algorithm which determines how to allocate user requests to the caches. Although simple algorithms can be used in practice (e.g. round-robin mechanism), we use linear programming to formulate the problem and compute the optimal request allocations. This allows us to focus on the content placement heuristics.

Several traditional traffic engineering (TE) metrics can be used to optimize the utilization of network resources. In this work we focus on minimizing the maximum link utilisation (max-u), which is commonly used in the TE literature [12] [13]. As shown in previous work, determining the optimal placement of contents that minimizes the max-u in the network is an NP-hard problem [14]. In this paper we investigate practical heuristics that can be used to solve this problem.

Resource management approaches usually rely on centralized solutions executed by a manager that has a global view of the current conditions (e.g. user demand, average link utilization). Although relatively simple to implement, centralized solutions have limitations in practice. In addition to the single point of failure issue, collecting global information about user demand and network conditions poses scalability problems since they can incur significant traffic overhead.

In this paper we propose an approach where, instead of relying on placement decisions received from a centralized manager, distributed cache managers coordinate among themselves to decide how to efficiently use the available caching space with the objective of better utilizing the network resources. To achieve this objective, cache managers are organized into an intelligent in-network substrate similar to the one we used previously for the purpose of adaptive resource management [15] [16]. The substrate is a logical structure used to facilitate the communication between the cache managers, which allows them to decide on reconfiguration actions in a coordinated manner. The frequency of content placement decisions is in the order of hours.

C. Modeling Choices

Several factors can affect the performance of a placement strategy, such as the user demand and the cache size. Selecting the strategy to apply is therefore a challenging issue. In this paper we focus on the influence of user demand characteristics on the performance of the proposed placement strategies. In order to limit the influence of other factors, we make the following modeling choices and assumptions.

We define the total volume of contents to cache in the network, V_X, as the sum of the size of each content x, so that $V_X = \sum_x s_x$. We also define the total caching space available in the network, V_M, as the sum of the capacity of each cache m, so that $V_M = \sum_m c_m$. By design, V_M is chosen so that each content can be stored in at least one of the network caches, i.e. $V_X \leq V_M$. In order to decorrelate the influence of the size of the contents, we assume that all the contents have the same size. In addition, we assume that all caches have the same capacity. The size of a cache at a particular location can depend for instance on several topological factors (e.g. node centrality). Determining the correlation between content size and user interest for the content, as well as taking into account different cache sizes, are challenging research issues but are outside the scope of this paper in which we simplify the problem formulation in order to limit the influence of other factors. We characterize the user demand by a) the total volume of requests for each content in the network (i.e. global content popularity - GCP), and b) the number of caching locations where each content is requested (i.e. geographic distribution of the interests - GDI).

1) Modeling the global popularity of the contents: Given that content popularity is long-tail distributed, it is commonly accepted that the GCP distribution can be represented by the Zipf law [17]. In this work, we use a Zipf distribution of parameter α (scaling-law coefficient) to model the GCP. Contents are indexed according to their rank r in the distribution, so that the global popularity of a content decreases when its index increases. In the rest of the paper, we identify each content x by its rank r, the values of the rank ranging from 1 to X.

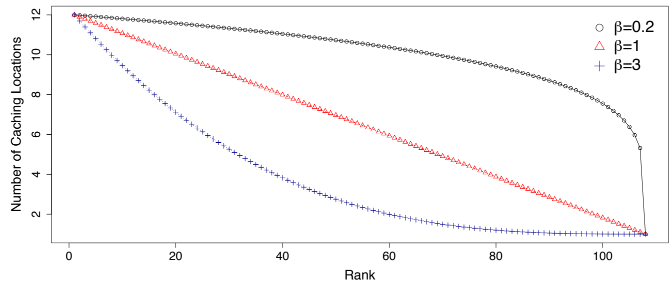

Fig. 2. Profile of the function f_β for three values of β

2) Modeling the geographical distribution of interests: The Zipf distribution does not account for the GDI. In the absence of real traces, we design a function that gives the number of caching locations in the network where each content r is requested. The proposed function is so that this number depends on the global popularity of the content in the network. Intuitively, the more popular a content is, the more likely it is to be requested at a large number of different locations. Very unpopular contents, on the other hand, are more likely to be requested from more specific geographical locations. With the proposed function, the minimum number of locations where a content can be requested is 1 and the maximum is the total number of caches in the network. In order to model various profiles of GDI (i.e. scenarios where most of the contents are requested from a large number of caching locations, or in contrast, from specific locations only), we introduce a parameter β in the function. The parameter β is a strictly positive constant that drives the characteristics of the GDI. The proposed function f_β is then defined as follows:

$$f_\beta(r) = 1 + (M-1)\left(\frac{(X-r)}{(X-1)}\right)^\beta \quad (1)$$

where M is the total number of caches in the network and X is the total number of contents in the list. The profile of the function f_β for three values of the parameter β is depicted in Fig.2 for $X = 108$ and $M = 12$.

As it can observed, when $0 < \beta < 1$, the function f_β exhibits a concave profile, i.e. most of the contents are requested from a large number of caches. The lower the value of β is, the higher the number of caching locations for each content is on average. The distribution of the interests is more homogeneous between the different caches, and the homogeneity increases as the value of β decreases. When $\beta = 1$, the function f_β is linear. The number of locations where each content r is requested linearly decreases with the global popularity rank of r. When $\beta > 1$, the function f_β exhibits a convex profile, i.e. most contents are requested from a subset of locations only. The higher the value of β is, the lower the number of caching locations for each content is on average. In this case, the GDI is more heterogeneous and the heterogeneity increases as the value of β increases. By tuning the value of the parameter β, different degrees of heterogeneity of the GDI can be modeled and as such, the function f_β can be used to represent a wide range of scenarios.

III. Placement Strategies

We propose two categories of content placement strategies that can be used by the cache managers in the access nodes. The proposed strategies satisfy the following constraints:

C1 There is at least one copy of each content cached in the network.

C2 A content r is copied in a cache only if there is enough caching capacity to accomodate r.

C3 A content r is copied in a cache only if r is not already cached at this location.

The strategies can be implemented in a decentralized fashion through the intelligent in-network substrate, which facilitates the exchange of information as explained in section II-B. More specifically, the objective of the proposed strategies is to determine the number of copies of each content to store in the network, as well as their placement at the different possible caching locations. The placement decisions are taken by the cache managers that coordinate among themselves to decide upon the most appropriate caching configurations to apply. The cache managers do not independently decide how to use their associated caching space (i.e. the contents stored locally), but instead they communicate through the substrate to exchange local information, such as the contents which are already stored locally. The two categories of strategies differ both in terms of the volume of information required and in terms of the characteristics of the information to be exchanged between the managers.

The decision-making mechanism to support the proposed placement strategies can follow an iterative approach as proposed for instance in [11]. In this case, content placement decisions are taken iteratively, so that only one cache manager is permitted to take a placement decision at a time. The order followed by the managers is pre-determined and is provided to each manager prior to the execution of the algorithm. For scalability purposes, a decision-making approach that can parallelize the decisions taken by each manager can also be considered [18]. In this case, however, a set of carefully selected constraints will need to be implemented at each cache manager to avoid inconsistent configurations. Due to space limitations, we do not elaborate on the implications of the decision-making mechanism on the two placement strategies proposed.

A. Local-Popularity driven Strategy

The local-popularity (LP) driven strategy follows a two-phase process. In the first phase, the cache managers collaborate to decide where to cache a first copy of each content ensuring that the first constraint is satisfied. In the second phase, each manager independently profits from the potential remaining caching space in the associated cache to replicate some of the more locally requested contents not already cached, until the total local space is consumed. Replicating contents locally limits the number of redirected requests. Each cache manager maintains the list $\mathcal{L}_{pop}$ of its locally requested contents ordered by decreasing local popularity (i.e. local aggregated volume of requests for each content). The list is provided to the managers at each re-configuration cycle.

First Phase: Each content placement decisions taken by a manager consists in selecting a single content item, from the locally requested ones, to cache locally, so that the selected content a) has the highest local popularity, b) is not already cached somewhere else in the network, and c) there is enough space to store the content. This procedure is executed until no further decision can be taken. In order to ensure that the first constraint is satisfied, a mechanism to check if all contents are cached is executed at the end of this phase. If not, non-cached contents are randomly placed in one of the caches with enough remaining capacity. This phase always terminates since, as explained in section II-C, $V_X \leq V_M$ by design.

Second Phase: Each manager selects a set of contents to replicate among the locally requested ones (i.e. from $\mathcal{L}_{pop}$), so that the selected contents are the locally most popular ones not already cached. The number of selected contents depends on the remaining caching space.

With this strategy the placement decisions taken by each manager depend essentially on the local user demand. Apart from the first phase, where information about contents previously cached in the network is exchanged/used, the managers do not use any network-wide information to decide upon local placements. Each new local placement is advertised to all other managers to ensure a consistent view of which contents are currently cached in the network.

B. Global-Popularity driven Strategy

The global-popularity (GP) driven strategy also follows a two-phase process. While the first phase ensures that the first constraint is satisfied, the second phase allows cache managers to replicate contents locally. Unlike the local popularity strategy however, collaborative decisions between the cache managers extend to the second phase. Each manager maintains a copy of the list of contents requested in the network ordered by decreasing global popularity, $\mathcal{L}_{pop}$ and for each content r, the list of caching locations where r is requested ordered by decreasing number of requests, $\mathcal{L}^r_{loc}$. As such, all managers have the same global knowledge about content popularity and geographical distribution of interests in the network. The lists are provided to the managers at each re-configuration cycle.

First Phase: The first phase of the GP strategy consists in iteratively considering each content r in the list $\mathcal{L}_{pop}$, so that to decide where place the first copy of r. The cache with the highest aggregated number of requests for r (and strictly greater than zero), and with sufficient space is selected. If none of the caches can satisfy these conditions, the content is temporarily disregarded and marked as *FAIL*. The procedure continues until all contents in $\mathcal{L}_{pop}$ are considered. Once the procedure is completed, caches with sufficient capacity are randomly selected to store any content marked as *FAIL*. In a similar fashion to the LP strategy, this phase always terminates since $V_X \leq V_M$ by design.

Second Phase: The procedure followed is similar to the one used in the first phase. The only difference lies in the number of caches selected to store a copy of each content. Instead of selecting one cache, a pre-defined number of n caches are selected, where n is at most equal to the total number of caches in the network. If it is not possible to find n caches satisfying the conditions, a copy of the content is stored in the maximum number of possible locations. The process stops when it is not possible to find a new feasible placement. The number n is an input of the algorithm and is provided to cache managers prior to the execution of the algorithm.

In this approach, placement decisions are driven by the global popularity of the contents. In addition, the algorithm tries to store each content r in the location where the demand for r is the highest. The number of copies of each content (i.e. replication degree) is driven by n. As n increases, the globally more popular contents tend to be more replicated than the others, whereas the replication degree of each content tends to be more uniform when n is small. The manager decisions are coordinated throughout the process. The communication overhead incurred by the GP strategy is therefore larger than that of the LP strategy.

IV. Evaluation

This section presents the evaluation of the performance of four strategies belonging to the two categories described in section III in terms of network resource utilization according to various user demand characteristics. The four strategies are the following:

1) The local-popularity driven strategy, noted LPS.
2) The global popularity driven strategy with replication factor $n = 1$, noted GPS_1.
3) The global popularity driven strategy with replication factor $n = 2$, noted GPS_2.
4) The global popularity driven strategy with replication factor $n = 3$, noted GPS_3.

A. Experiment Settings

We use the PoP-level Abilene network [19] to simulate the proposed caching infrastructure presented in Fig.1. The Abilene network has 12 nodes (which are all source nodes) and 30 unidirectionnal links. In our simulation we associate one cache to each of the network nodes and assign the same capacity to all links. The total caching space is set so that twice the total volume of contents to cache can be accomodated in the network, and is uniformely distributed between the 12 caching locations.

In the absence of real demand traces, we generate synthetic user demand profiles. Each profile is characterized by a pair (α, β), as explained in section II-C. In order to evaluate a wide range of profiles, we consider 20 values of parameter α with $\alpha \in [0.1; 2]$, and 20 values of parameter β with $\beta \in [0.2; 10]$, i.e. a total of 400 (α, β) pairs. Each pair defines the total number of requests for each content in the network, and for each content m, the number of locations where m is requested. We generate 100 samples for each pair, i.e. we randomly select different locations from where each content is requested, and as a result, we obtain a total of 40,000 evaluation samples. The total volume of requests in the network is constant in all experiments. We consider a list of approximately 100 content items to cache in the network for the evaluation to be manageable in time.

B. Evaluation Methodology

In order to simulate various user demand profiles, we use different values of parameters α and β. In practice, however, characterizing the demand through α and β may not be feasible, since it may be difficult to retrieve the actual value of the parameters given the monitored information in the network. We define two metrics that can be used, instead, to characterize the user demand.

1) Metric H_G: The metric H_G describes the heterogeneity of the geographic distribution of interests for the different contents. We note d_{mr} the demand for content r at caching location m. We note M the total number of caches in the network, and X the total number of contents. For each content r, we define a parameter $\bar{h}_g(r)$ that characterizes the heterogeneity in terms of geographical distribution of interests (GDI) for content r in the network:

$$\bar{h}_g(r) = \frac{\sqrt{\frac{1}{M}\sum_m \left(d_{mr} - \mathbb{E}_m(d_{mr})\right)^2}}{\sum_m d_{mr}} . \tag{2}$$

where $\mathbb{E}_m(d_{mr})$ is the expectation of the demand for content r at each caching location m, i.e. $\mathbb{E}_m(d_{mr}) = \frac{1}{M}\sum_m d_{mr}$. The parameter $\bar{h}_g(r)$ represents the standard deviation of the demand for content r at each cache m normalized by the total demand for r in the network.

We then define H_G as the normalized sum of the $\bar{h}_g(r)$:

$$H_G = \frac{1}{X}\sum_r \bar{h}_g(r) . \tag{3}$$

2) Metric H_P: To characterize the heterogeneity in terms of global popularity between the different contents in the network, we define the metric H_P as follows:

$$H_P = \frac{\sqrt{\frac{1}{X}\sum_r \left(\sum_m d_{mr} - \mathbb{E}\left(\sum_m d_{mr}\right)\right)^2}}{\sum_r \sum_m d_{mr}} . \tag{4}$$

where $\mathbb{E}\left(\sum_m d_{mr}\right)$ is the expectation of the total number of requests for a content in the network, i.e. $\mathbb{E}\left(\sum_m d_{mr}\right) = \frac{1}{X}\sum_r \sum_m d_{mr}$ and $\sum_r \sum_m d_{mr}$ is the total volume of requests in the network. Therefore, parameter H_P represents the standard deviation of the total number of requests for each content in the network normalized by the total volume of requests in the network.

As it can be noticed, the value of α and β is not required to determine the value of H_G and H_P. These are obtained through the knowledge of the total number of contents to cache, the total number of caches in the network and the demand for each content at each caching location only. These can be available in the network, either in the form of static parameters (e.g. number of caches), or through monitoring information (e.g. aggregated user demand at the network edges). In order to observe the relationship between the two metrics and the parameters α and β, we plot in Fig.3 and Fig.4 the values of H_P and H_G obtained for each of our evaluation samples against the different values of α and β.

As it can be observed in the figures, H_P is correlated to parameter α but is independent of parameter β. On the contrary, H_G is independent of parameter α but is correlated to parameter β. More specifically, the value of H_P strictly increases with the value of α, and, as such, a larger value of H_P represents a higher degree of heterogeneity between the global popularity of contents in the network. In a similar fashion, the value of H_G strictly increases with the value of β, and, as such, a larger value of H_G represents a higher degree of geographic heterogeneity (i.e. GDI) in the network. It is worth mentioning that although there exists a correlation between the values of H_P and H_G and the parameters α and β, the formulae used to determine the actual values of the two metrics are independent of α and β. As such, these metrics are not adapted to the current settings only, which makes them valuable to analyze the popularity characteristics in any content distribution scenario. In addition, these can be directly applied to the monitored data since only the demand for each content is required to compute the values of these metrics.

C. Results and Analysis

We compare the performance of each of the four strategies in terms of the maximum link utilization (max-u) resulted in the network for different demand profiles. A demand profile is characterized by a pair of values (H_P;H_G). The best strategy is the one that results in the lowest max-u.

Fig.5 shows the results for each of the 40,000 samples. For readability purposes, the figure is divided in four windows, one for each strategy. For each sample we determine the best strategy by comparing the different max-u. Each point in each strategy window means that this is the best strategy for the relevant evaluation sample. It is important to note that according to our experiment settings, each pair is associated to 100 samples, and as such, one point in a graph may actually represent several samples. As it can be observed from the figure, the performance of each strategy follows different trends. In the region defined by the smallest values of H_P and H_G, strategy GPS_1 is exclusively the best strategy. On the contrary, in the region defined by a value of H_P greater than 0.25 and value of H_G smaller than 0.7, the best strategy is always strategy LPS. For other values of H_P and H_G, however, determining which strategy can achieve the best performance depends on the sample considered, and as such, no strategy can be exclusively identified as the best strategy.

Due to the replication logic, all contents tend to have a similar replication degree with strategy GPS_1. As it can be observed from the results, such a strategy is preferable when both the GCP and the GDI are more homogeneous, i.e. when the values of H_P and H_G are low. In this case, using a strategy that partitions the overall caching space in a fair manner between the contents can lead to better network resource utilization. Compared to strategy GPS_1, the content replication degree is less uniform between the contents with strategy GPS_2 and GPS_3. Given that the speed at which the overall caching space is consumed is driven by the factor n, as defined in section III-B, globally popular contents have a significantly higher replication degree than other contents, the difference increasing as factor n increases. When the value of H_P is low but the value of H_G increases (i.e. the GDI becomes more heterogeneous), a placement strategy that behaves in more discriminating fashion in terms of replication degree achieves better performance. Due to the absence of coordination during the second phase of strategy LPS, globally popular contents tend to be significantly more replicated than the others. As it can be observed, for large values of H_P, a strategy that aggressively increases the replication degree of the most popular contents leads to better network resource utilization. In this case, the volume of traffic incurred in the network is mainly dominated by the requests for these contents. By replicating these contents in most locations, a large volume of traffic can thus be removed from the network, which leads to lower max-u.

In order to see how the performance of the different strategies compares to each other, we analyze the average deviation of the max-u obtained by each strategy with the minimum max-u reported for each (H_P,H_G) pair. The average deviation is determined by computing, for each strategy, the deviation from the minimum for each sample of a given (H_P,H_G) and by averaging the results over the total number of samples considered. For each strategy, we depict the values obtained in a density graph, as shown in Fig.6. The magnitude of the deviation is represented by different shades of grey, the darkest shade corresponding to a deviation of 0% and the lightest to the maximum deviation observed in the experiments. Given that the values obtained spanned over a large range, we use a logarithmic scale for measuring the deviation. As it can be observed, strategy LPS performs uniformly well for H_P greater than 0.3 and H_G smaller than 0.7 since the average deviation is close to 0. In contrast, the three other strategies obtain poor performance on average in that region. Very good performance is also obtained by strategy LPS for low values of H_P and large values of H_G. It can also be noticed that strategy GPS_1 performs uniformly well for the lowest values of H_P and $H_G < 0.5$. Whereas the results in Fig.5 indicate that the lowest max-u can be achieved with strategies GPS_2 and GPS_3 in some cases, it can be observed in Fig.6 that these strategies lead on average to much higher resource utilization. The performance is therefore less predictable.

The performance of each strategy in terms of resource utilization is influenced by the user demand. More precisely, the evaluation results show that, in some cases, knowing H_P and H_G is not enough to determine which strategy to apply

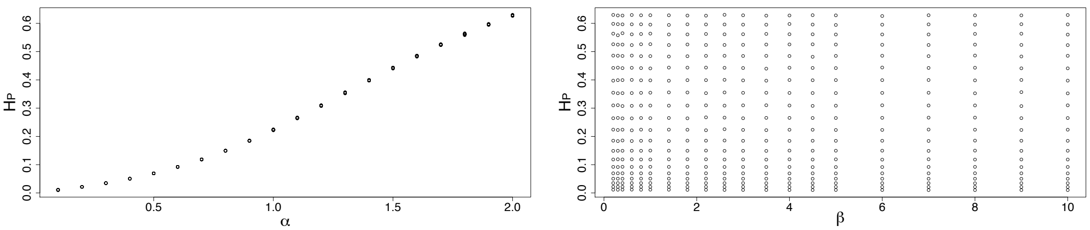

Fig. 3. Value of H_P vs. parameter α and parameter β

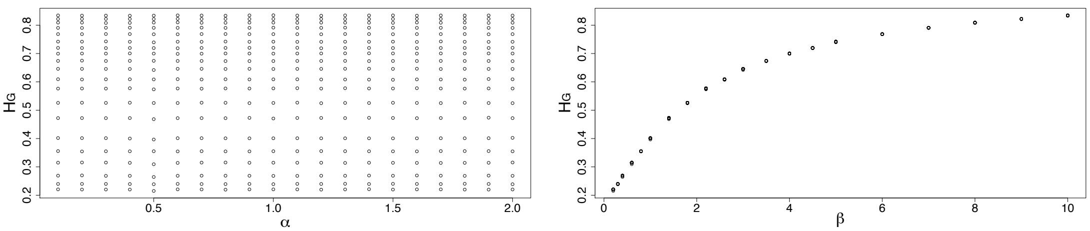

Fig. 4. Value of H_G vs. parameter α and parameter β

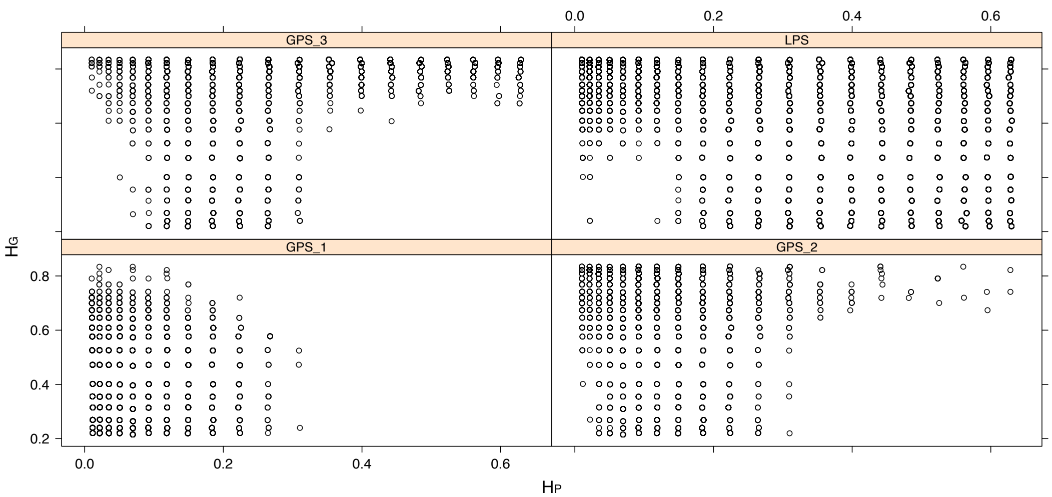

Fig. 5. Performance of the four strategies according to different values of H_P and H_G

since the best performance can be achieved by several of them. It is possible, however for certain values of H_P and H_G, to identify one strategy that outperforms the others. As such, the proposed metrics can provide useful indications regarding the performance one strategy can achieve over another and, as such, can be used by the ISP to improve the utilization of network resources.

V. Related Work

The placement problem has received a lot of attention from the research community over the years and has been addressed in different contexts. For instance [20] investigates the problem for the selection of the most appropriate physical locations to place web server replicas in order to minimize network cost.

Closer to our work are the approaches proposed in [21] [18] [22] and [11], which all focussed on intelligent techniques to replicate content across different network locations in order to better utilize network resources. Previous work, such as [23], has also investigated specific solutions for hierarchical network infrastructures, which have been applied in the context of IPTV [24] [22]. A distributed placement strategy in the context of distributed replication groups is proposed in [18], where nodes in the group cooperate to take replication decisions that can minimize the overall network cost. In that work, the authors assume the existence of a constant and uniform cost for retrieving any content within the group, and, as such, they do not take into account network characteristics such as link utilisation. The content placement problem has also been tackled from the point of view of game theory in [25] under the conditions of infinite cache capacity. An autonomic cache management framework for information-centric networks

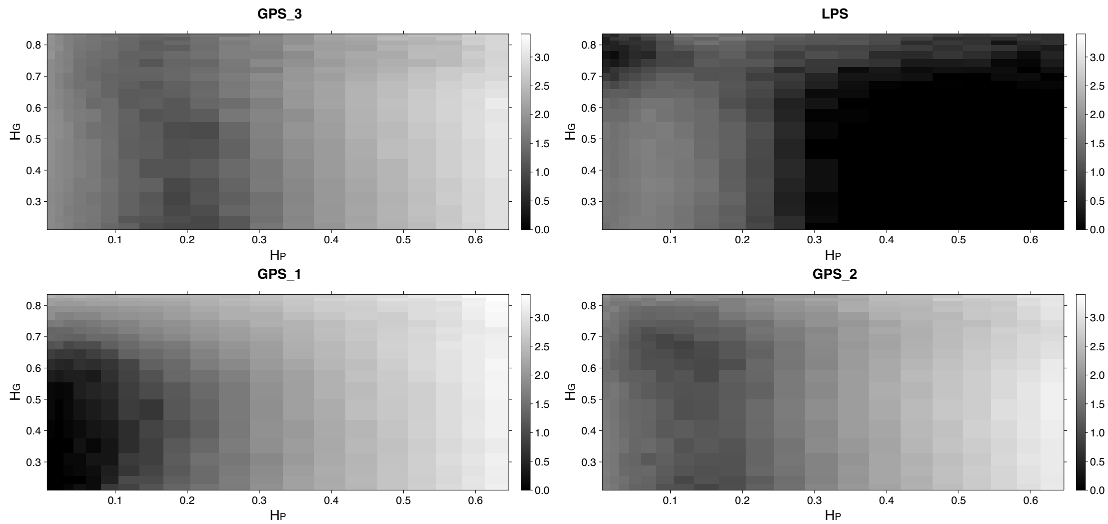

Fig. 6. Density graphs of the average deviation of max-u for each strategy

was presented in [11]. Three dynamic distributed replacement algorithms with different levels of cooperation between the caches and complexity were evaluated. These could complement our approach by optimizing the initial content placement configuration according to the short-term dynamics of user demand. Optimal solution structures for the combined problem of object placement and assignment of requests to caching locations (i.e. server selection) were investigated in [26], [27] and [14], where the objective was to optimize the global network cost. Although the proposed algorithms can provide near-optimal solutions to the coupled optimization problem, the approaches have limitations in practice. They assume, in particular, the availability of global information about user demand and network conditions at a centralized location, which may incur a significant overhead and can have scalability limitations. In addition, given that the two problems are solved concurrently, the solution can take a long time to converge.

Given the significant impact that content delivery has on the utilization of ISP network, some work has recently started to investigate new models and frameworks to support the interaction between ISPs and CDNs [4] [5] [9] [10]. In [4], the authors highlight that CDN providers and ISPs can indirectly influence each other, by performing server selection and traffic engineering operations respectively, and they investigate different models of cooperation between the two entities. In [5], the authors propose a framework to support joint decisions between a CDN and an ISP with respect to the server selection process. This framework allows the ISP and the CDN to collaborate by exchanging some local information (network utilization from the ISP side and server conditions from the CDN side), so that it can result in better control of the resources. In contrast to these approaches, our solution focuses on empowering ISPs with caching capabilities, which can allow them to implement their own content placement and server selection strategies. ISP-centric caching approaches have also been considered in [9] and [10]. However, the full-blown CDN service to be supported by an ISP, as proposed in these approaches, can incur high operational costs, given that ISPs will have to maintain large storage capacities, and may thus be an economically unviable solution.

VI. Summary and Future Work

In this paper we propose a cache management scheme by which simple and lightweight content placement strategies can be used by an ISP to determine the placement of content in the various network caching points according to the characteristics of user demand. Although the proposed ISP caching solution may not provide the level of service that CDNs do, it is low-cost compared to traditional CDN services which (i) require a lot of power for operating and cooling the storage infrastructure, (ii) have collocation costs, and (iii) require human supervision. In contrast, our approach is automated, has smaller distributed storage, and has simplified functionality that can be realized by commodity hardware boxes. The deployment of such an approach can allow ISPs to manage their network resources more effectively. This is can subsequently improve user experience which could work towards the benefit of some CDN providers.

In future extension of this work, we plan to investigate the influence of different content and cache sizes on the performance of the proposed strategies. We also plan to apply the proposed scenario to different network topologies. This work assumes that content popularity can be determined by an ISP and subsequently provided as input to the proposed placement algorithms. In practice, however, determining this information without input from the CDN is a key issue which we plan to investigate. More generally, the cooperation model between the ISP and CDN providers is an issue that relates to the evolution of CDNs and the increased participation of ISPs in this and we will both follow relevant developments in this area and also investigate models for the interaction between the two.

REFERENCES

[1] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind akamai (travelocity-based detouring)," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 435–446, aug 2006.

[2] Akamai CDN. [Online]. Available: http://www.akamai.com

[3] Limelight Networks CDN. [Online]. Available: http://www.limelight.com

[4] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative content distribution and traffic engineering in an ISP network," in *Proc. SIGMETRICS '09*, Seattle, WA, USA, 2009, pp. 239–250.

[5] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann, "Content-aware traffic engineering," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 413–414, jun. 2012.

[6] Content Delivery Networks Interconnection (CDNI). [Online]. Available: http://tools.ietf.org/wg/cdni/draft-ietf-cdni-framework/

[7] Coral CDN. [Online]. Available: http://www.coralcdn.org

[8] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "Donar: decentralized server selection for cloud services," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 231–242, aug. 2010.

[9] N. Kamiyama, T. Mori, R. Kawahara, S. Harada, and H. Hasegawa, "ISP-operated CDN," in *Proc. of the 28th IEEE international conference on Computer Communications Workshops*, ser. INFOCOM'09, 2009, pp. 49–54.

[10] K. Cho, H. Jung, M. Lee, D. Ko, T. Kwon, and Y. Choi, "How can an ISP merge with a CDN?" *IEEE Commun. Mag.*, vol. 49, no. 10, pp. 156–162, oct. 2011.

[11] V. Sourlas, P. Flegkas, L. Gkatzikis, and L. Tassiulas, "Autonomic cache management in information-centric networks," in *Proc. IEEE Network Operations and Management Symposium (NOMS'12)*, apr. 2012, pp. 121–129.

[12] S. Balon, F. Skivée, and G. Leduc, "How well do traffic engineering objective functions meet te requirements?" in *Proc. IFIP-TC6 NETWORKING'06*, 2006, pp. 75–86.

[13] N. Wang, K. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for internet traffic engineering," *IEEE Communications Surveys Tutorials*, vol. 10, no. 1, pp. 36–56, quarter 2008.

[14] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale vod system," in *Proc. Co-NEXT '10*, Philadelphia, Pennsylvania, 2010, pp. 1–12.

[15] D. Tuncer, M. Charalambides, G. Pavlou, and N. Wang, "Dacorm: A coordinated, decentralized and adaptive network resource management scheme," in *Proc. IEEE Network Operations and Management Symposium (NOMS'12)*, apr. 2012, pp. 417–425.

[16] M. Charalambides, G. Pavlou, P. Flegkas, N. Wang, and D. Tuncer, "Managing the future internet through intelligent in-network substrates," *IEEE Netw.*, vol. 25, no. 6, pp. 34–40, nov. 2011.

[17] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," Telecom ParisTech, Tech. Rep., 2011.

[18] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Distributed selfish replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 12, pp. 1401–1413, dec. 2006.

[19] The Abilene Network. [Online]. Available: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/

[20] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *Proc. IEEE INFOCOM'01*, vol. 3, 2001, pp. 1587–1596.

[21] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," vol. 25, no. 4, mar. 2002, pp. 376–383.

[22] J. Dai, Z. Hu, B. Li, J. Liu, and B. Li, "Collaborative hierarchical caching with dynamic request routing for massive content distribution," in *Proc. IEEE INFOCOM'12*, mar. 2012, pp. 2444–2452.

[23] M. Korupolu and M. Dahlin, "Coordinated placement and replacement for large-scale distributed caches," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 6, pp. 1317–1329, nov./dec. 2002.

[24] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM'10*, mar. 2010, pp. 1–9.

[25] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, "Selfish caching in distributed systems: a game-theoretic analysis," in *Proc. PODC '04*, St. John's, Newfoundland, Canada, 2004, pp. 21–30.

[26] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems," *SIAM J. Comput.*, vol. 38, no. 4, pp. 1411–1429, aug. 2008.

[27] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Comput. Oper. Res.*, vol. 35, no. 12, pp. 3860–3884, dec. 2008.