Evaluating the Utility of Content Delivery Networks

Konstantinos Stamos
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
kstamos@csd.auth.gr

George Pallis
Department of Computer Science
University of Cyprus
Nicosia, Cyprus
gpallis@cs.ucy.ac.cy

Athena Vakali
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
avakali@csd.auth.gr

Marios D. Dikaiakos
Department of Computer Science
University of Cyprus
Nicosia, Cyprus
mdd@cs.ucy.ac.cy

ABSTRACT

Content Delivery Networks (CDNs) balance costs and quality in services related to content delivery. This has urged many Web entrepreneurs to make contracts with CDNs. In the literature, a wide range of techniques has been developed, implemented and standardized for improving the performance of CDNs. The ultimate goal of all the approaches is to improve the utility of CDN surrogate servers. In this paper we define a metric which measures the utility of CDN surrogate servers, called *CDN utility*. This metric captures the traffic activity in a CDN, expressing the usefulness of surrogate servers in terms of data circulation in the network. Through an extensive simulation testbed, we identify the parameters that affect the CDN utility in such infrastructures. We evaluate the utility of surrogate servers under various parameters and provide insightful comments.

Categories and Subject Descriptors: C.2.4 [Computer Communication Networks]: Distributed Systems

General Terms: Experimentation, Performance

Keywords:CDN pricing, Content Delivery Networks, network utility

1. INTRODUCTION

Content Delivery Networks (CDNs) [28] have gained considerable attention in the past few years. A CDN is an overlay network across Internet, which consists of a set of surrogate servers distributed around the world, routers and network elements. The surrogate servers, which are deployed in multiple locations, cooperate with each other, transparently moving content in the background to optimize the end user experience. When a client makes a request, the CDN generally chooses a surrogate server at a location that is near the client, thereby optimizing the perceived end-user experience. An indicative CDN is depicted in Figure 1. Detailed information about CDN mechanisms are presented in [23, 28].

Motivation. CDNs play a key role in the Internet infrastructure since their high end-user performance and cost savings have urged many Web entrepreneurs to make contracts with CDNs [16]. Nowadays, there are many commercial CDNs, including Akamai, AT&T, Limelight and Mirror Image. In a recent study [11], authors quantitatively evaluate the performance of two commercial large-scale CDNs (Akamai and Limelight) with respect to the number of surrogate servers, their internal DNS designs, the geographical locations of their surrogate servers and their DNS and surrogate server delays. The authors provide an extensive background research and insightful comments. Except of commercial CDNs, there are also a number of non-commercial ones [5, 6]. CDNs continuously become more competitive by offering novel services to the public. The development of a new service usually includes high investments. The most traditional CDN services include distributing static Web pages and large file downloads, such as software patches. CDNs also provide application acceleration, supporting e-commerce and delivering dynamic content, back-end databases and Web 2.0 applications. CDNs are also assisting enterprise customers in providing rich Web applications with context and location-aware services. Leading CDN companies such as Akamai and Limelight are now offering streaming media delivery, distributing media for CNN, BBC, and so on. The enormously popular user-generated video site, YouTube, is currently distributed by the Limelight CDN.

In order to be able to offer all the above services to the public, several technical issues should be considered. Specifically, critical decisions should be taken related to CDN framework setup, content distribution and management, and request management approaches. In the literature, a wide range of techniques [4, 13] has been developed, implemented and standardized for improving the performance of CDNs. The ultimate goal of all the approaches is to improve the utility of CDN surrogate servers. Towards this direction

UPGRADE-CN'09, June 9, 2009, Munich, Germany.

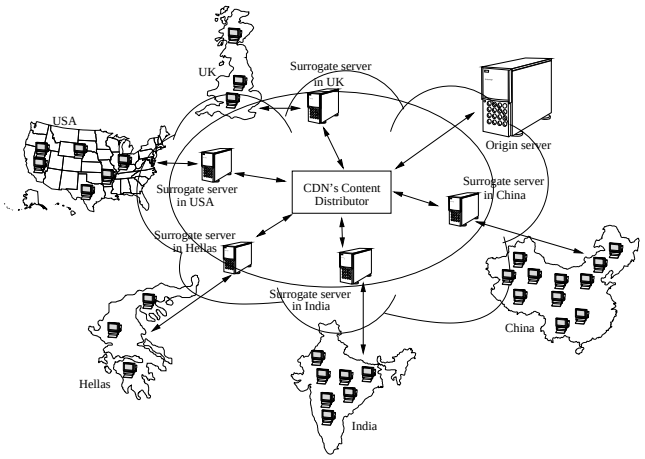

Figure 1: A typical Content Delivery Network.

the authors in [9] describe the design and development of a content-serving utility that provides highly scalable Web content distribution over the Internet.

Contribution. In this work, we evaluate the utility of surrogate servers in different policies using the notion of net utility. In particular, net utility is a value that expresses the relation between the number of bytes of the served content against the number of bytes of the pulled content (from origin or other surrogate servers). Also, we use the notion of net utility in order to define a CDN pricing policy. Given the vast number of competing CDN providers, it is essential to determine the optimal pricing for CDN services. In general, the pricing of CDNs is a complex problem. Subscribing content providers can be highly heterogeneous in terms of their traffic patterns and the type of content they handle [7]. At the same time, the CDN providers have to announce a single pricing policy that accounts for all these different traffic types. Through an extensive simulation testbed, we identify the parameters that affect the net utility in CDN infrastructures. Using a wide range of Web sites, this study reveals several observations about the utility of surrogate servers and provides incentives for their exploitation in the design of CDNs. Up to now little attention has been paid to evaluate the CDN utility. To the best of our knowledge, this work is one of the first efforts to make an extensive analysis of the parameters that affect the utility of surrogate servers in a CDN infrastructure.

Roadmap. The remainder of this paper is organized as follows: Section 2 presents the earlier recent research work on CDNs. Section 3 describes the CDN utility metric as well as how this metric contributes to defining a pricing model for CDNs. Section 4 presents the simulation testbed and Section 5 the experimentation results. Section 6 concludes the paper.

2. RELATED WORK

CDNs have gained considerable attention in the past few years. The earlier recent research work on CDNs can be divided into the following four major categories:

- **Establishing theoretical models:** Theoretical models can be used to efficiently solve the resource allocation and management problems in a CDN [2]. In particular, mathematical models have been proposed in the literature to address several issues related to where to locate surrogate servers [22], which content to outsource [12], evaluating pricing models [10] and request routing mechanisms [1, 19]. Mathematical modeling techniques can also be used to gain insight to a variety of CDN problems arising in practice and to determine what mitigating actions can be taken. For instance, the authors of [18] use a Lagrangian-based solution algorithm based on a mathematical model to evaluate the effect of data clustering on the total revenue of a CDN provider using this algorithm. Moreover, theoretical models facilitate the solution of CDN problems by providing a generic framework on which efficient exact solution algorithms can be devised. These are also used as benchmarks to assess a variety of heuristic methods [14]. However, all these models deal with the individual problems separately, without taking into account possible interplays between them. Therefore, while they provide valuable information, the need for simulations is not tackled where all those problems can be aggregated.

- **Developing policies for CDN infrastructure:** Several issues are involved in CDNs since there are different decisions related to content distribution and management approaches. These can be summarized as follows: a) surrogate servers placement [15, 22], b) content outsourcing and delivery [4, 13, 25], c) cache organization in surrogate servers [14, 26] and d) p2p and Grid technologies for the development of novel CDNs [8].

- **Developing academic CDNs:** Instead of delegating the content delivery to a commercial CDN provider, the Web content servers participate in an academic CDN with low fees. Academic CDNs are real world systems and run in a wide area environment, the actual Internet topology. A well-known academic CDN, Globule [21], is an open source CDN which is operated by end-users. The Web content servers participate in the Globule by adding a module to their Apache server. Another academic CDN is the CoralCDN [6]. In order to use the CoralCDN, the Web content servers, which participate in this network, append .nyud.net:8080 to the hostname in a URL. Through DNS redirection, the clients with unmodified Web browsers are transparently redirected to nearby CORAL surrogate servers. Another well-known academic CDN is the CoDeeN [5]. In order to use the CoDeeN, as previously, a prefix must be added to the hostname in a URL. Regarding the academic performance of CDNs, they offer less aggregate storage capacity than commercial CDNs and, require wide adoption of the system to bring substantial performance benefits to the end-users.

- **Developing simulation testbed systems:** This category deals with developing a CDN simulation system, which will simulate a dedicated set of machines to reliably and efficiently distribute content to clients on behalf of the origin server. Such a testbed runs locally on a single machine and contrary to the academic CDNs it is a simulated environment. An analytic simulation tool for CDNs, called CDNsim, has been developed

in [27]. CDNsim has been designated to provide a realistic simulation for CDNs, simulating the surrogate servers, the TCP/IP protocol and the main CDN functions.

3. CDN UTILITY

Net utility is a value that expresses the relation between the number of bytes of the served content against the number of bytes of the pulled content (from origin or other surrogate servers). A similar metric has also been used in [17] for a p2p system. It is bounded to the range [0..1] and provides an indication about the CDN traffic activity.

Formally, we quantify a net utility u_i of a CDN surrogate server i by using the following equation:

$$u_i = \frac{2}{\pi} \times arctan(\xi) \quad (1)$$

The intuition of this metric is that a surrogate server is considered to be useful (high net utility) if it uploads content more than it downloads, and vice versa. The parameter ξ is the ratio of the uploaded bytes to the downloaded bytes. The resulting net utility ranges to [0..1]. The value $u_i = 1$ is achieved if the surrogate server uploads only content ($\xi = infinity$). On the contrary, the value 0 is achieved if the surrogate server downloads only content. In the case of equal upload and download, the resulting value is 0.5. Finally, the CDN utility u can be expressed as the mean of the individual utilities of each surrogate server. Considering that a CDN has N surrogate servers, the CDN utility u can be defined as follows:

$$u = \frac{\sum_{i=1}^{N} u_i}{N} \quad (2)$$

The notion of CDN utility can also be used as a parameter to CDN pricing policy. Typically, a CDN outsources content on behalf of content provider and charges according to a usage (traffic) based pricing function. The ultimate goal is to identify the final cost for the content provider under a CDN infrastructure. Since net utility measure captures the CDN usage, the respective net utility of a CDN can be easily translated into a price for its offered services.

In this context, we integrate the notion of net utility in the pricing model that has been presented in [10]. In particular, the monetary cost of the Web content provider under a CDN infrastructure is determined by the following equation:

$$U_{CDN} = V(X) + \tau(N) \times X - C_o - P(u) \quad (3)$$

where U_{CDN} is the final cost of Web content provider under a CDN infrastructure, $V(X)$ is the benefit of the content provider by responding to the whole request volume X, $\tau(N)$ is the benefit per request from faster content delivery through a geographically distributed set of N CDN surrogate servers, C_o is cost of outsourcing content delivery, $P(u)$ is the usage-based pricing function, and u is the CDN utility. As we will show, the CDN utility u is tightly related to the usage/traffic imposed and therefore can be applied to $P(u)$. In the rest of the paper we refrain from using an exact monetary definition of the cost. Instead, we focus on how u and $P(u)$ are affected.

4. SIMULATION TESTBED

The CDN providers are real-time applications and they are not always available for research purposes. Therefore, for the evaluation purposes, it is crucial to have a simulation testbed for the CDN functionalities and the Internet topology. Furthermore, we need a collection of Web users traces which log access to a Web server's content through a CDN. In order to identify visiting sessions in the requests we need as well the structure of the content, i.e. the structure of a Web site. Although we can find several users traces on the Web, real traces from CDN providers are not available to us nor the respective Web sites. Thus, we are faced to use artificial data. Moreover, the use of artificial data enables us to evaluate extensively which parameters affect the CDN utility. In fact, we are able to perform simulations under common parameter values as identified in the literature [20] and also under extreme values. As such, we can establish in an empiric way the theoretical performance limits of the CDN. In this framework, we have developed a full simulation environment, which includes the following:

- a system model simulating the CDN infrastructure,
- a network topology generator,
- a Web server content generator, modeling file sizes etc.,
- a client request stream generator capturing the main characteristics of Web users' behavior.

4.1 CDN model

To evaluate the CDN utility measure, we used our complete simulation environment, called CDNsim, which simulates a main CDN infrastructure. The full source code be found at http://oswinds.csd.auth.gr/~cdnsim/. It is based on the OMNeT++ library[1] which provides a discrete event simulation environment. All CDN networking issues, like surrogate server selection, propagation, queuing, bottlenecks and processing delays are computed dynamically via CDNsim, which provides a detailed implementation of the TCP/IP protocol, implementing packet switching, packet retransmission upon misses etc.

We consider a CDN with 100 surrogate servers which have been located all over the world. Each surrogate server in CDNsim is configured to support 1000 simultaneous connections. The default cache capacity of each surrogate server has been defined as a percentage of the total size in bytes of the Web content provider's Web site. We also consider that each surrogate server cache is updated using a standard LRU cache replacement policy. Experiments have shown that the cache size has direct impact to the performance of a CDN [4, 13]. Usually a larger cache results in less requests to be redirected to another surrogate server (in a cooperative p2p CDN) or to the origin server.

4.2 Network topology

In a CDN topology we may identify the following network elements: surrogate servers, origin server (Web content provider's main server), routers and clients. Additionally, we consider the existence of a Tracker. The Tracker is considered as a server, belonging to the CDN, which is responsible to redirect the requests to the appropriate surrogate/origin

[1]http://www.omnetpp.org/article.php?story=20080208111358100

server. It is aware of the content of each server and of the network topology.

The routers form the network backbone where the rest of the network elements are attached. The distribution of servers and clients in the network affects the performance of the CDN. Different network backbone types result in different "neighborhoods" of the network elements. Therefore, the redirection of the requests and ultimately the distribution of the content is affected. In our testbeds we use four different network backbone flavors: AS, Waxman, Transit stub and Random. Each of them contains 3037, 1000, 1008 and 1000 routers respectively. The routers retransmit network packets using the TCP/IP protocol between the clients and the CDN. All the network phenomena such as bottlenecks and network delays, and packet routing protocols are simulated. Note that the AS Internet topology with a total of 3037 nodes captures a realistic Internet topology by using BGP routing data collected from a set of 7 geographically-dispersed BGP peers. Finally, in order to minimize the side effects due to intense network traffic, we assume a high performance network with 1 Gbps link speed.

4.3 Web server content generation

In order to generate the Web server content we developed a tool which produces synthetic but realistic Web site graphs. We considered many of the patterns found in real world Web sites such as zipfian distributions for sizes [20]. According to Zipf's law, a Web site contains mostly relatively small objects following the zipfian distribution. The distribution is modified by a parameter z which affects the slope of the distribution. The higher the z is the steeper the slope of the distribution is and vice versa. For instance, if $z = 0$ then all objects have the same size, whereas, if $z = 1$ then the size of the objects fades exponentially. By default we have generated Web sites with 50000 objects of $1GB$ total size and $z = 1$.

4.4 Requests generation

As far as the requests stream generation is concerned, we used a generator, which reflects quite well the real users access patterns. Specifically, this generator, given a Web site graph, generates transactions as sequences of page traversals (random walks) upon the site graph [20]. We are focused especially on the following parameters as more representative to affect the CDN utility:

- *Popularity distribution.* As it is observed in [20], the pages popularity of a Web site follows Zipfian distribution. According to Zipf's law, the higher the value of z is the smaller portion of objects covers the majority of the requests. For instance, if $z = 0$ then all the objects have equal probability to be requested. If $z = 1$ then the probability of the objects fade exponentially. The popularity distribution affects the CDN utility as we will show later. Therefore, we used the range 0, 0.5 and 1 for z in order to capture an average case and more extreme ones.

- *Popularity-size correlation of objects.* In a Web site, different objects exhibit different demand by the clients [3]. The popularity of an object expresses the probability of a client request to request this specific object. Specifically, the correlation between object popularity and size ranges in $[-1..1]$. Positive correlation indicates that the larger objects are more popular than the smaller ones. On the contrary, negative correlation indicates that the smaller objects are more popular than the larger ones. A zero correlation suggests that the size and popularity are irrelevant to each other. This correlation as we will show in the next section affects the CDN utility. In particular, we examine the CDN utility under the values of 0, 1, -1 for the correlation to capture the average and the extreme cases.

In this work, we have generated 1 million users requests. We consider that the requests arrive according to an exponential distribution with mean interarrival time equal to $1sec$. Then, the Web users requests are assigned to CDN surrogate servers taking into account the network proximity, which a the typical way followed by CDNs providers. In this context, we examine the following CDN redirection policies:

- *Closest surrogate server with cooperation (closest ss \w coop):* A client performs a request for an object. The request is redirected transparently to the closest surrogate server A in terms of network topology distance. The surrogate server uploads the object, if it is stored in its cache. Elsewhere, the request is redirected to the closest to A surrogate server B, that contains the object. Then, the surrogate server A downloads the object from B and places it in its cache (some content may be removed in this step according to the cache replacement policy). If the object is not outsourced by the CDN, the surrogate server A downloads the object from the origin server directly. Finally the object is uploaded to the client.

- *Closest surrogate server without cooperation (closest ss \wo coop):* This policy follows the same redirection mechanism with the previous one. The major difference is that if the surrogate server A is unable to satisfy the request, it is not redirected to another surrogate server. Instead, the surrogate server A downloads the object directly from the origin server.

- *Random surrogate server with cooperation (random ss \w coop):* The requests are distributed randomly without using any proximity metric. The positive characteristic of this policy is the load balancing since the requests are distributed evenly among the surrogate servers. However, the network traffic is increased because the object transfers are performed via long network paths. We use in order to identify the performance bounds of the p2p cooperation.

5. EVALUATION

In this section, we study which parameters affect the CDN utility. In this context, we have performed four sets of experiments. The first set studies the impact of the network topology backbone to CDN utility. The second set examines the CDN utility under various popularity distributions while the third one tests its impact regarding the correlation between objects popularity and objects size. Finally, the fourth set of experiments studies the CDN utility under various CDN redirection policies. The summary of the parameters used in the four experimentation sets is presented in Table 1.

Parameter	Experimentation 1	Experimentation 2	Experimentation 3	Experimentation 4
Web site size	$1GB$	$1GB$	$1GB$	$1GB$
Web site number of objects	50000	50000	50000	50000
Web site z for size	1	1	1	1
Correlation size vs. popularity	0	0	0,−1,1	0
Number of requests	1000000	1000000	1000000	1000000
Mean interarrival time of the requests	$1sec$	$1sec$	$1sec$	$1sec$
Distribution of the interarrival time	exponential	exponential	exponential	exponential
Requests stream z	0.5	0.5,1.0,0.0	0.5	0.5
Link speed	$1Gbps$	$1Gbps$	$1Gbps$	$1Gbps$
Network topology backbone type	AS, Waxman, Transit stub, Random	AS	AS	AS
Number of routers in network backbone	3037, 1000, 1008, 1000	3037	3037	3037
Number of surrogate servers	100	100	100	100
Number of client groups	100	100	100	100
Number of content providers	1	1	1	1
Cache size percentage of the Web site's size	2.5%, 5%, 10%, 20%, 40% and 80%	2.5%, 5%, 10%, 20%, 40% and 80%	2.5%, 5%, 10%, 20%, 40% and 80%	2.5%, 5%, 10%, 20%, 40% and 80%
CDN redirection	*closest ss \w coop*	*closest ss \w coop*	*closest ss \w coop*	closest ss \w coop, closest ss \wo coop, *rand ss \w coop*

Table 1: Summary of simulations parameters

5.1 Evaluation measures

We evaluate the performance of CDN under regular traffic. It should be noted that for all the experiments we have a warm-up phase for the surrogate servers' caches. The purpose of the warm-up phase is to allow the surrogate servers' caches to reach some level of stability and it is not evaluated. The measures used in the experiments are considered to be the most indicative ones for performance evaluation. Specifically, the following measures are used:

CDN utility: It is the mean of the individual net utilities of each surrogate server in a CDN. The net utility (defined in equation 3) is the normalized ratio of uploaded bytes to downloaded bytes. Thus, the CDN utility ranges in [0..1]. Using the notion of CDN utility we express the traffic activity in the entire CDN. Values over 0.5 indicate that the CDN uploads more content than it downloads through cooperation with other surrogate servers or the origin server. In particular, for the uploaded bytes we consider the content uploaded to the clients and to the surrogate servers. Values lower than 0.5 are not expected in the CDN schemes. The value 0.5 is an extreme case where each request is a non-outsourced object.

Hit ratio: It is the ratio of requests that has been served, without cooperation with other surrogate servers or the origin server, to the total number of requests. It ranges in [0..1]. High values of hit ratio are desired since they lead to reduced response times and reduced cooperation. Usually the hit ratio is improved by increasing the cache size and it is affected by the cache replacement policy.

Byte hit ratio: It is the hit ratio expressed in bytes. It is a more representative metric for measuring bandwidth consumption and network activity, especially when there is positive or negative correlation between size and popularity in a Web site. It ranges in [0..1].

Mean response time: It is the mean of the serving times of the requests to the clients. This metric expresses the clients experience by the use of CDN. Lower values indicate fast served content.

5.2 CDN utility vs. Network topology

Simulation setup. The examined Web site includes 50000 objects of total size $1GB$. The correlation of the size with the popularity is set to 0. Additionally, we generated the respective request stream that contains 1000000 requests with $z = 0.5$. The cache size for each surrogate server is set to 2.5%, 5%, 10%, 20%, 40% and 80% of the total Web site size. For instance, the extreme case of 100% cache size means that all the content of Web server content has been fully mirrored to all the CDN surrogate servers. The network topology, since it is our examined parameter, is set to AS, Waxman, Transit stub and Random.

Discussion. Figure 2 depicts the CDN utility evolution for different cache sizes and network topologies. Axis x represents the percentage of surrogate server cache size with respect to the total Web site size while the y axis represents the CDN utility. Each line refers to a different network topology. The most notable observation is that there is a single peak in the performance of the CDN utility at 10% cache size. This peak, although it has different values, occurs at the same cache percentage for all network topologies. Therefore, the performance peak, in terms, of CDN utility, is invariant of the network topology.

Giving more insight to the performance peak, we should identify what happens to the eras before and after the peak. Before the peak, the cache size is quite small. Few replicas are outsourced to the CDN and the surrogate servers fail to

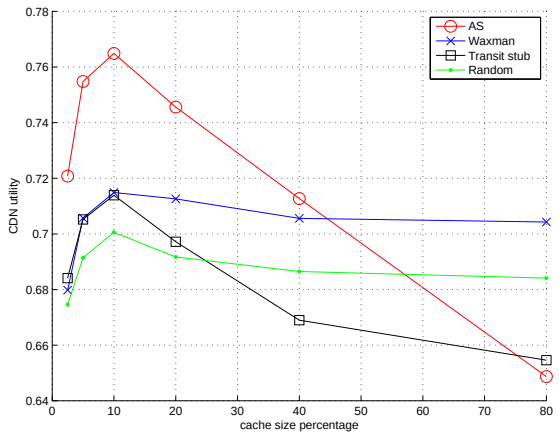

Figure 2: CDN utility vs. Network topology

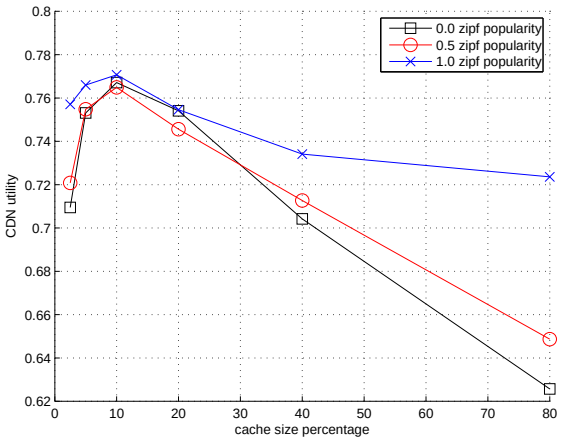

Figure 3: CDN utility vs. Popularity distribution

cooperate. Most of the requests refer to objects that are not outsourced at all. Consequently, the surrogate servers refer to the origin server in order to gain copies of these objects. This leads to poor CDN utility as the surrogate servers upload less content. As the cache size increases, the amount of replicated content in the CDN increases as well. Therefore, the cooperation among the surrogate servers is now feasible and thus the CDN utility increases. After the peak, the cache size is large enough; consequently, this results in reducing the cooperation among the surrogate servers.

If we set the cache size to 100% and run the simulation for a very long time period, each cache of surrogate servers will have replicated all the content of the Web site (full mirroring). Therefore, after a very long time period ($\lim_{t\to\infty} u(t) = 1$) the CDN utility will approximate 1. This is the theoretical limit of CDN utility. However, in practice this will never happen since the content of the Web sites changes through time leading to cache misses.

The network topology makes no difference to the performance peak. However, the individual CDN utilities are quite different. The Random and the Waxman network topologies demonstrate a flat evolution of CDN utility suggesting poor distribution of the requests in CDN. On the other hand, the AS and the Transit stub exhibit more steep slopes to the performance suggesting a more intense phenomenon, as described previously, about the two eras.

Conclusions. In this experiment we have observed the evolution of CDN utility against increasing cache size. We have shown that *there is a performance peak in terms of CDN utility at a certain cache size, which is invariant under different network topologies.* Considering that the capacity allocation in surrogate servers affects the pricing of CDN providers (see equation 3) we view this finding as particular important. This provides an indication about the optimal size of site that can be replicated to surrogate servers. *Replicating a small size of Web site content, the observed performance peak guarantees satisfactory performance by reducing the traffic to origin server.*

5.3 CDN utility vs. Popularity distribution

Simulation setup. The same synthetic Web site was used as in the previous set. We have generated three request streams that contain 1000000 requests. We examine the z parameter for the popularity by setting it to 0, 0.5 and 1. The higher the z the smaller the percentage of the objects that have large popularity. The cache size for each surrogate server is set to 2.5%, 5%, 10%, 20%, 40% and 80%. As previous, the cache size is defined as the percentage of the total bytes of the Web server content. The network topology is fixed to AS.

Discussion. Figure 3 illustrates the CDN utility under different cache sizes and popularity distributions. More specifically, the x axis represents the percentage of surrogate server cache size with respect to the total Web site size while the y axis represents the CDN utility. Each line refers to a different popularity distribution. We study the behavior of CDN utility in conjunction with the hit ratio as recorded in Figure 4.

There are two main observations in this set:

- For different popularity distributions we observe the same performance peak at the same cache size. This is an indication that the CDN requires to allocate a minimum cache size for each Web site in order to enable effective cooperation among its surrogate servers.
- Higher values of z result in higher CDN utility. This is expected since as z increases only a small portion of the objects absorb a very high percentage of the requests. This is also supported in Figure 4 where for $z = 0$ the objects are uniformly requested and the hit ratio is very poor even for large caches. In this case, the cache is unable to "learn" the request stream. For $z = 1$ the hit ratio is very high even for very low cache sizes since a small portion of objects are requested.

Conclusions. The key question we investigate here is how the popularity affects pricing in terms of CDN utility. Our study is able to provide an answer to this: If we consider only the maximum CDN utility as criterion, then *the popularity does not interfere with pricing.*

5.4 CDN utility vs. Size distribution

Simulation setup. For this experiment, three data sets with 50000 objects were generated where each one has total

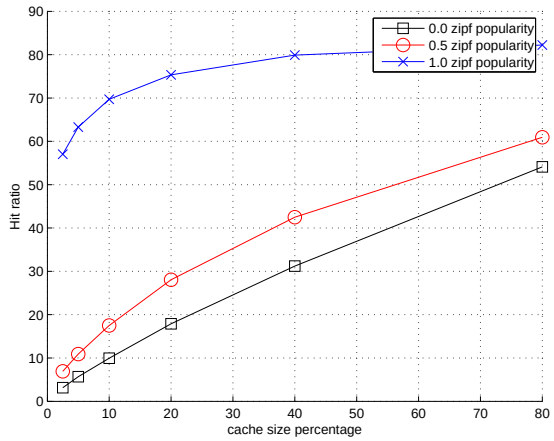

Figure 4: Popularity distribution vs. Hit ratio

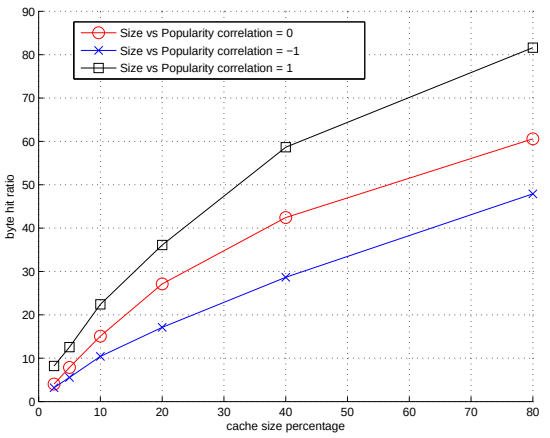

Figure 6: Size distribution vs. Byte hit ratio

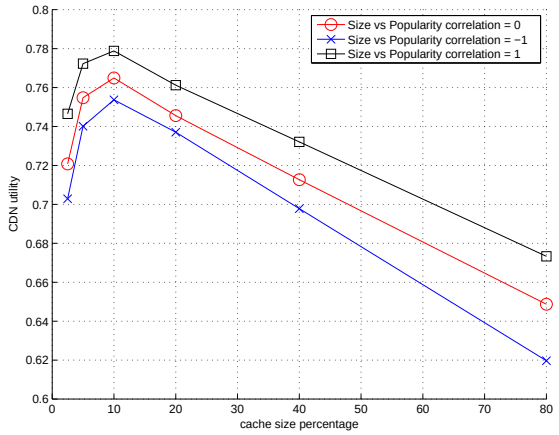

Figure 5: CDN utility vs. Size distribution

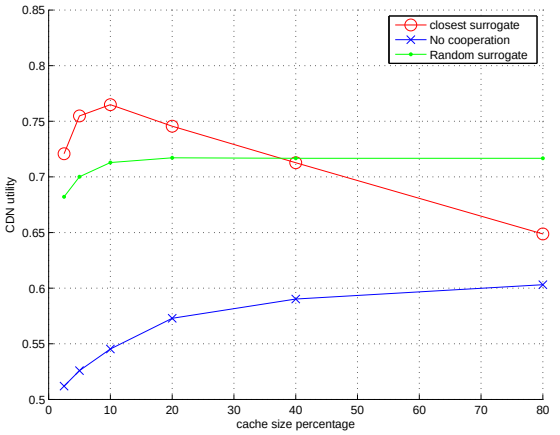

Figure 7: CDN utility vs. CDN redirection policy

size $1GB$. The Zipf parameter z of the popularity of objects is set to 1. Their difference lies on the correlation of the objects size with respect to popularity. We used the extreme cases of correlation -1, 0 and 1. Positive correlation results in large objects to be requested with higher probability. Negative correlation suggests that small objects are more popular while the zero correlation does not favor either size. Additionally, we generated the respective request stream that contains 1000000 requests with $z = 0.5$. The cache size for each surrogate server is set to 2.5%, 5%, 10%, 20%, 40% and 80% of the total Web site size. Finally, the network topology is set to AS.

Discussion. The CDN utility is recorded in Figure 5. The x axis is the percentage of surrogate server cache size with respect to the total Web site size, the y axis is the CDN utility and the different lines refer to different popularity correlations. We examine this data set by taking into account the byte hit ratio metric as presented in Figure 6. There are two primary observations:

- Regardless the popularity correlation, the CDN utility peak exists at the same cache size percentage.
- The positive correlation enhances the CDN utility. This behavior is expected since more large objects are being transferred. The worst CDN utility is observed at the negative correlation where the small objects are favored. The zero correlation lies in between. These observations are supported by the byte hit ratio metric in Figure 6. Positive correlation leads to very high byte hit ratio while negative leads to very poor.

Conclusions. The question that remains to be answered is how the size vs. popularity correlation affects the CDN utility and ultimately the pricing. According to the above findings, we may conclude that *large files trafficking is in favor to content provider*. This leads to increased CDN utility and thus monetary cost reduction (see equation 3) for the Web content provider.

5.5 CDN utility vs. CDN redirection policy

Simulation setup. In this set we used the same synthetic Web site and requests stream as in the first set. Using the AS network topology we examine the CDN utility under the three CDN redirection policies.

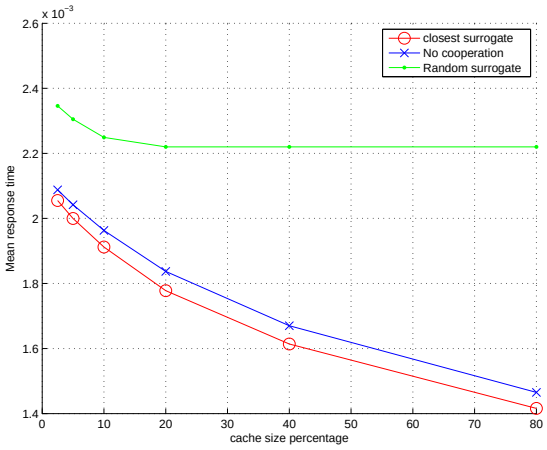

Figure 8: Mean response time vs. CDN redirection

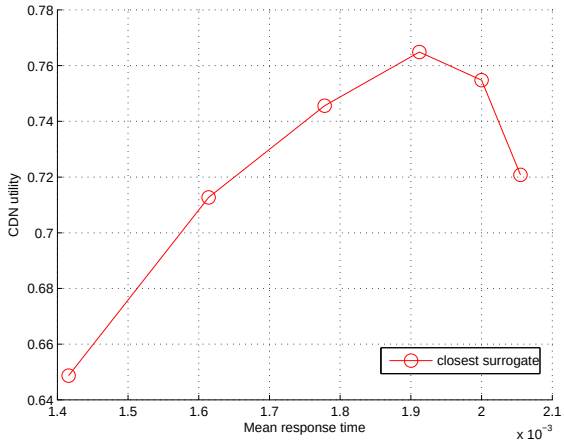

Figure 9: CDN utility vs. Mean response time

Discussion. Figures 7 and 8 record the CDN utility and mean response times of the requests. The x axis represents the surrogate servers cache size. It is defined as the percentage of the total bytes of the Web server content. The y is the CDN utility and mean response time respectively. The individual lines represent the different CDN redirection policies. To begin with, the CDN utility in the case of the *closest surrogate server with cooperation* exhibits a performance peak. The *closest surrogate without cooperation* does not exhibit such a peak. This is expected since there is no cooperation. The amount of uploaded content is affected solely by each individual surrogate server performance. Increasing cache size leads to increasing cache performance. In the case of "random surrogate server with cooperation" the CDN utility after the peak leads to a plateau. This occurs due the fact that the requests are randomly distributed in the CDN. Therefore the cache replacement algorithm is unable to "fit" to the request stream.

Another important metric for CDNs evaluation is the mean response time. This measure captures the users satisfaction. Figure 8 depicts the mean response time with respect to cache size for different redirection policies. The best performance is achieved by the *closest surrogate server with cooperation*, demonstrating the superiority of the p2p cooperation. In the case of *random surrogate server with cooperation* the mean response times are quite poor since there is high internetwork traffic. This is caused due to the random distribution of the requests without taking into account any network proximity criterion. Instead of using such a naive method a CDN could be beneficial even if there is no cooperation among surrogate servers. This is evident in the case of *closest surrogate without cooperation*. Although the results in this case are satisfactory, this does not hold true in high traffic phenomena such as the flash crowd events [24].

Figure 9 examines in greater depth the relation between CDN utility and mean response time. From this figure we observe that *the highest CDN utility value results in low mean response time but not in the lowest that can be achieved.* When we have the lowest mean response time (this means that the Web site has been fully replicated to all the surrogate servers), the CDN utility is low since the surrogate servers do not cooperate with each other.

Conclusions. To sum up, an interesting question here is to identify how the CDN redirection scheme affects the monetary cost for a content provider. Our study is able to provide an answer to this: *We observed that a poorly designed redirection policy would not exhibit the desired CDN utility peak.*

6. CONCLUSION

In this work we examined how the CDN utility is affected by various parameters. More specifically we have evaluated the CDN utility under different network topologies, traffic models and Web site models. The observations are quite enlightening. In particular the primary contributions of this work were:

- A definition of CDN utility as a metric to capture the traffic activity in a CDN is defined. The CDN utility expresses the usefulness of a CDN in terms of data circulation in the network.
- Insightful commentary via extensive experimentation is provided, upon different parameters affecting the CDN utility (with or without p2p cooperation).
- A performance peak, in terms of CDN utility has been detected. The peak is invariant of the network topology, the traffic model and the Web site model.
- The problem of selecting the optimal content size that should be replicated in surrogate servers is addressed by taking into consideration the CDN utility metric.
- The CDN utility has been considered as a parameter for defining a CDN pricing policy.

The experimentation results are quite encouraging to spawn a set of possible future works. In fact, it is necessary to examine the CDN utility under more parameters and configurations. Potentially interesting results may occur during a flash crowd event. Finally, we strongly believe that the CDN utility can be considered as a way to measure the "health" of a CDN and to consider an advanced pricing model.

7. REFERENCES

[1] T. Bektas, J.-F. Cordeau, E. Erkut, and G. Laporte. Exact algorithms for the joint object placement and request routing problem in content distribution networks. *Computers and Operations Research*, 35(12):3860–3884, 2008.

[2] T. Bektas and I. Ouveysi. *Lecture Notes in Electrical Engineering*, volume 9, chapter Mathematical Models for Resource Management and Allocation in CDNs, pages 225–250. Springer Berlin Heidelberg, July 2008.

[3] M. Busari and C. Williamson. Prowgen: a synthetic workload generation tool for simulation evaluation of web proxy caches. *Comput. Netw.*, 38(6):779–794, 2002.

[4] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. H. Katz. Efficient and adaptive Web replication using content clustering. *IEEE Selected Areas in Communications*, 21(6):979 – 994, August 2003.

[5] CoDeeN. CoDeeN : A CDN for PlanetLab. http://codeen.cs.princeton.edu.

[6] CORAL. CORAL CDN. http://www.coralcdn.org.

[7] M. D. Dikaiakos and A. Stassopoulou. Content-selection strategies for the periodic prefetching of www resources via satellite. *Computer Communications*, 24(1):93–104, June 2001.

[8] G. Fortino and C. Mastroianni. Enhancing content networks with p2p, grid and agent technologies. *Future Generation Computer Systems*, 24(3):177–179, 2008. Special Issue Editorial.

[9] P. Gayek, R. Nesbitt, H. Pearthree, A. Shaikh, and B. Snitzer. A web content serving utility. *IBM Syst. J.*, 43(1):43–63, 2004.

[10] K. Hosanagar, J. Chuang, R. Krishnan, and M. Smith. Service adoption and pricing of Content Delivery Network (CDN) services. Technical report, Social Science Research Network, 2006.

[11] C. Huang, A. Wang, J. Li, and K. Ross. Measuring and evaluating large-scale cdns. In *IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 15–29, New York, NY, USA, 2008. ACM.

[12] J. Kangasharju, J. Roberts, and K. W. Ross. Object replication strategies in content distribution networks. *Computer Communications*, 25(4):367 – 383, March 2002.

[13] D. Katsaros, G. Pallis, K. Stamos, A. Vakali, A. Sidiropoulos, and Y. Manolopoulos. CDNs Content Outsourcing via Generalized Communities. *IEEE Transactions on Knowledge and Data Engineering*, 21(1), 2009.

[14] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis. On the Optimization of Storage Capacity Allocation for Content Distribution. *Computer Networks*, 47:409–428, 2005.

[15] B. Li, X. Deng, M. J. Golin, and K. Sohraby. On the optimal placement of web proxies in the internet: The linear topology. In *HPN '98: Proceedings of the IFIP TC-6 Eigth International Conference on High Performance Networking*, pages 485–495, Deventer, The Netherlands, The Netherlands, 1998. Kluwer, B.V.

[16] Market. Content Delivery Networks, Market Strategies and Forecasts (2001-2006). Technical report, AccuStream iMedia Research, 2006.

[17] B. Mortazavi and G. Kesidis. Model and simulation study of a peer-to-peer game with a reputation-based incentive mechanism. In *Information Theory and Applications Workshop*, 2006.

[18] T. V. Nguyen, F. Safaei, P. Boustead, and C. T. Chou. Provisioning overlay distribution networks. *Comput. Netw.*, 49(1):103–118, 2005.

[19] C. A. S. Oliveira and P. M. Pardalos. A survey of combinatorial optimization problems in multicast routing. *Comput. Oper. Res.*, 32(8):1953–1981, 2005.

[20] V. N. Padmanabhan and L. Qiu. The content and access dynamics of a busy web site: findings and implications. In *SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 111–123, New York, NY, USA, 2000. ACM.

[21] G. Pierre and M. Steen. Globule: a collaborative content delivery network. *IEEE Communications Magazine*, 44(8):127–133, August 2006.

[22] L. Qiu, N. V. Padmanabhan, and M. G. Voelker. On the Placement of Web Server Replicas. Technical report, Ithaca, NY, USA, 2001.

[23] M. Rabinovich and O. Spatsheck. *Web caching and replication*. Addison Wesley, 2002.

[24] P. Ramamurthy, V. Sekar, A. Akella, B. Krishnamurthy, and A. Shaikh. Using mini-flash crowds to infer resource constraints in remote web servers. In *INM '07: Proceedings of the 2007 SIGCOMM workshop on Internet network management*, pages 250–255, New York, NY, USA, 2007. ACM.

[25] A. Sidiropoulos, G. Pallis, D. Katsaros, K. Stamos, A. Vakali, and Y. Manolopoulos. Prefetching in Content Distribution Networks via Web Communities Identification and Outsourcing. *World Wide Web Journal*, 11(1):39–70, March accepted 2008.

[26] K. Stamos, G. Pallis, C. Thomos, and A. Vakali. A similarity-based approach for integrated Web caching and content replication in CDNs. In *Proceedings of 10th International Database Engineering and Applications Symposium*, pages 239–242, December 2006.

[27] K. Stamos, G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos, and Y. Manolopoulos. CDNsim: A Simulation Tool for Content Distribution Networks. *ACM Transactions on Modeling and Computer Simulation*, 2009.

[28] A. Vakali and G. Pallis. Content delivery networks: Status and trends. *IEEE Internet Computing*, 7(6):68–74, November 2003.