Evaluating User-perceived Benefits of Content Distribution Networks

Claudia Canali
University of Parma
claudia@weblab.ing.unimo.it

Valeria Cardellini
University of Roma "Tor Vergata"
cardellini@ing.uniroma2.it

Michele Colajanni
University of Modena
colajanni@unimo.it

Riccardo Lancellotti
University of Modena
lancellotti.riccardo@unimore.it

Abstract

Content Distribution Networks (CDNs) are a class of successful architectures that the most popular Web sites use to enhance their performance. The basic idea is to address Internet bottleneck issues by replicating and caching the content of the customer Web sites and to serve it from the edge of the network. In this paper we evaluate to what extent the use of a CDN can improve the response time perceived from the users. We consider a large set of different scenarios, with particular regard to different network conditions and client connections, that have not been considered in previous performance evaluations. We conclude that CDNs can offer significative performance gain in normal network conditions, but the advantage of using CDNs can be reduced by heavy network traffic. Moreover, we found that, if CDN usage is not carefully designed, the achieved speedup can be suboptimal.

Keywords: Content delivery, Caching, End-to-end performance, Edge servers.

1 INTRODUCTION

In a scenario where client requests have to reach the origin server and the responses must travel backwards, many network and server bottlenecks may affect performance of Web content delivery.

There are three main possible network bottlenecks. The so called *first mile* that is, the network link between the origin Web server and the Internet, can become congested if it is under-provisioned with respect to the traffic of requests that is typically bursty and subject to flash crowds. The *last mile* that is, the connection between the end user and the Internet, is another known source of performance problems. Moreover, most HTTP request/response has to traverse many Autonomous Systems because of the fine grain distribution of the Internet architecture (it consists of more than 9000 Autonomous Systems, where even the most important network providers handle less than 5% of the global traffic). The *peering points* among Autonomous Systems is another critical point for end-to-end Web performance because they are seldom over-sized due to large costs that few providers are interested to or can afford.

It is also worth to observe that end-to-end performance does not depend on the network only. The server side is another possible bottleneck, especially in the last years when the percentage of personalized and dynamically generated Web content is continuously increasing. There are two opposite approaches to face the performance problems of Web content delivery.

In the *core model*, a Web cluster consisting of locally distributed servers [5] can solve most problems related to the server side, but it is ineffective with respect to the network-related issues. Moreover, the connection of the Web cluster to the Internet may easily become the bottleneck (first mile problem).

On the other hand, the *network edge model* aims to a complete or partial replication of the Web site content over geographically distributed servers. There are two main approaches to the so called *edge delivery*: the distributed architecture of multiple servers is managed by the content provider or it is delegated to a third party. The former solution can be convenient for a Web site that has a permanent popularity, even although a minority of content providers can afford the complexity of setting up and managing a geographically distributed architecture. As a consequence, the latter seems the most viable solution especially when the Web site has to deal with flash crowds and short periods of intensive traffic (e.g., a Web commerce site in December, a sport site during the event). This outsourcing alternative has created a new market of so called *Content Distribution Network* (CDN) companies. Many have appeared (and disappeared), and now two or three share the largest part of the market [13]. The largest companies (i.e., Akamai [1, 7], Mirror Image [16], Speedera [20]) provide an infrastructure of thousands of geographically distributed Web farms (called *surrogate* or *edge servers*), most of which are placed at the edge of the Internet that is, at the points of presence of the most important ISP.

The basic philosophy of these architectures is to improve performance by cooperative pro-active caching. With respect to traditional proxy caches [17], a CDN solution can achieve much higher cache hit rates thanks to several mechanisms. The CDNs must not deal with all Web content, because their working set is limited to the content of the customer Web sites. Moreover, the CDN edge servers work in cooperation

with the origin servers, hence they can use mechanisms that are typical of the reverse proxy technology, such as prefetching, push caching, and consistency control of replicated resources.

The main limitation of CDN services is due to their costs that are still expensive. Hence, it is of key importance to give independent evaluations of the CDN companies claims. Our purpose is to denote the space where this outsourcing solution provides real performance benefits to the end user.

To this purpose, we use a new tool (called **CDNperf**) that analyzes and compares the user-perceived response time of content delivery achieved with and without the use of CDNs. Our study considers a large set of network and system conditions during different periods, covering in all a length of time of almost two years. To the best of our knowledge, no other studies analyzed CDNs performance for a so long time. This allows us to provide some original insights to the performance of CDN-based delivery. Moreover, this paper integrates and extends some recent work that is most closely related to our study, especially [3, 12, 13], as better explained in section 3.

The work by Krishnamurthy et al. [13] represents to now the most extensive study on CDN performance evaluation. The authors modify httperf (a widely used tool for Web server benchmarking) to measure the performance of various CDN companies. Our work extends theirs from a different perspective. First, we employ the real Web pages encountered in actual Web sites, not using a canonical Web page. Second, we evaluate the benefits deriving by the introduction of CDNs comparing the page response time measured when using or not the CDN service. Third, we examine the contribution of DNS lookup time to the total page download time.

The Medusa proxy tool [3, 12] has been used to evaluate the performance of CDNs limited to the Akamai company. While their studies looks at Web delivery systems, we compare different CDN architectures and various client-to-Internet connections. Our network-oriented focus considering different Internet traffic conditions, different client locations, different last mile bandwidths extends the analysis of CDNs performance to a wider range of parameters.

In another interesting paper, that is more distant from our studies, the authors have compared the performance of CDNs against traditional Web delivery and peer-to-peer file sharing systems [18]. Some other performance studies have been carried out through simulation. For example, Kangasharju et al. show that the retrieval of objects on the same Web page from multiple servers may cause a performance degradation [10]. However, due to the complexity of the real infrastructure to be modeled [14], we believe that analytical and simulation techniques are well suited to evaluate new research ideas, such as the design of new CDN policies and mechanisms, rather than to analyze the performance of existing CDN architectures.

The rest of this paper is organized as following. Section 2 provides an overview of the main routing and delivery mechanisms that are adopted by CDN architectures. Section 3 describes the evaluation methodology we used to collect and process our performance data. Section 4 discusses the main features of the tool that we implemented and used for collecting the results for a large set of different scenarios. Section 5 presents our study on a significant set of Web sites that use a CDN architecture to deliver their contents. Section 6 concludes the paper with some final remarks.

2 Routing and delivery mechanisms

In this section we review the main phases involved in the content service from the CDN infrastructure to the consumers. We identify three core phases, namely the *selection* phase to determine the edge server/s that is/are considered best suited to respond, the *request routing* phase in which a proper mechanism is used to direct the client request to the target edge server(s), and the *delivery* phase during which the requested content is transferred to the client.

The selection and request routing phases may be interleaved, as a CDN can adopt some complex routing mechanism acting at different network levels to select one or more dispatching entities and to direct a client request to the "best" edge server(s). In fact, although a CDN has a number of possibilities, our study demonstrates that the server selection phase typically chooses the server that is "nearest" to the client that has issued the request to the customer Web site. The evaluation of the proximity among clients and edge servers is usually a function of network topology and dynamic link characteristics, although some interesting issues are still open about the concept of Internet proximity [15, 19]. It is likely that CDNs apply even some more sophisticated server selection algorithms taking into account other system factors, such as server availability and utilization. However, in all our analyses the consequences of these algorithms never emerged. Server selection and routing mechanisms provided quite stable results, probably because during the tests no critical events affected any edge servers.

As a main focus of this paper is to evaluate the performance impact of CDN architectures at the network level, it is important to outline the main routing mechanisms that can be used to direct a client request to an edge server. To this purpose, it is important to recall the typical components of a Web document. A user issues one request at a time for a *Web page*, which is intended to be rendered to the user as a single unit. Nevertheless, each user request (click) causes multiple client-server interactions because a Web page is typically a multi-part document consisting of a collection of objects. The most common example of Web page consists of a base HTML file describing the page layout and a number of embedded objects referenced by the base HTML file. All CDN servers deliver static files, although some of them are

specialized to serve even dynamic content. In this case, the content can be dynamically assembled on some edge server, for example by means of the recent Edge Side Includes technology [8].

The most used request routing mechanisms in CDNs can be mainly divided into the classes of DNS-based and application-layer mechanisms [4]. The use of the authoritative DNS server of the Web site as the request dispatcher has been initially proposed for locally and geographically distributed Web-server systems (in the latter architecture for the first level routing) [5, 6]. In these systems, the DNS server maps the site hostname to the IP address of one server node. Similarly, the same technique can be used to direct client requests to the CDN edge servers. In this case, the authoritative DNS server of the origin site delegates to the modified authoritative DNS server of the CDN company the resolution for those hostnames whose content is delivered through the CDN. In [13] this approach is referred to as *full-site content delivery*, being the origin server completely hidden to clients. Besides the well-known DNS limitation on request control due to the address caching mechanisms, the full-site content delivery approach has the drawback of a coarse-grained, content-blind routing decision [6, 17]. Moreover, as in the full-site content delivery approach the origin server is completely hidden by the authoritative DNS server to whom the content service has been delegated, it is not possible to compare from a client point of view the performance gain achieved by CDNs.

Therefore, in this paper we decide to focus on the performance evaluation of the alternative class of architectures, where CDNs use DNS-based routing in combination with some application-layer routing mechanism. Another motivation is that this scheme has a wider spread usage (mainly due to the still large popularity of Akamai) and a major flexibility with respect to the full-site content delivery approach. In this scheme, called *partial-site content delivery* [13], a client request first reaches the origin server that then applies one of the two most common application-layer routing mechanisms to redirect the requests to an edge server:

HTTP redirection. The redirection mechanism provided by the HTTP protocol allows an origin server to respond to a client request with a 301 or 302 status code in the response header. These codes instruct the client to resubmit its request to another node.

URL rewriting. Through URL rewriting, the origin server changes dynamically the links for the embedded objects within the requested Web page so that they point to another node. In such a way, the base page is returned by the origin server and all (or most) embedded objects are served by some other node(s).

External analyses have demonstrated that none of these two mechanisms clearly outperforms the other, as both of them add an extra round-trip time to the request processing. However, our analysis verified that all considered Web sites use URL rewriting. Hence, our focus is on this routing mechanism. Figure 1 shows a service request flow in a CDN adopting the partial-site content delivery approach, when the client retrieves the container page from the origin server (step 1) and embedded objects from the edge server (step 3) to which the client has been redirected. We also include the CDN's DNS server (step 2) that may play a role in the request routing process as explained below.

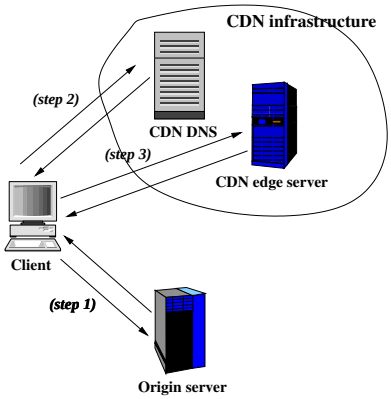


Figure 1: A request flow in a CDN based on partial-site content delivery.

The URL rewriting mechanism is typically used in combination with DNS-based routing [7, 17]. For URL rewriting the hostnames of the embedded object URLs are rewritten to the hostnames resolved in a subsequent step by the CDN's DNS server. With HTTP redirection, the hostnames of the CDN's DNS server are provided within the Location header of the HTTP response. In both instances, the CDN's DNS infrastructures maps the edge server name to a corresponding IP address, possibly by using multiple tiers of proprietary name servers combined with very low Time-To-Live values. For example, this is the scheme implemented by Akamai [7, 11].

In alternative, the DNS interposition could be avoided by having the origin server rewrite the IP addresses of the edge servers instead of their hostnames. This last solution avoids the need for multiple DNS lookups to serve the objects contained in a Web page, but makes less flexible and more visible the entire architecture, hence it is seldom used (we did not find any site using this URL-IP rewriting scheme). In some recent works it has been observed that the multiple DNS resolution required by the URL-hostname rewriting scheme is likely to increase the response time perceived by the users [13, 19]. One goal of our performance study is to verify under which conditions this hypothesis is valid.

The last phase completing the service from a CDN infrastructure is the *content delivery*, during which all the objects composing the requested Web page are transferred to the client. The number of entities that carry out the delivery clearly depends on the selection phase and the request routing mechanism adopted by the CDN. Under the partial-site content delivery approach, the origin server provides to the user the container page, while the embedded objects are delivered by one or more edge servers that have been selected by the dispatching entity(ies). In our performance analysis we investigate to what extent the number of edge servers employed by a CDN to serve the Web page may affect the performance perceived by the users.

3 Evaluation methodology

In this section we present the main network and system features that we consider noteworthy in order to evaluate the real benefits of CDN on user-perceived performance. We also explain how our methodology and analysis integrate previous work. In Section 4, we describe the main features of the tool CDNPerf that implements the main step of the evaluation methodology.

The performance gain of CDNs has not been widely studied due to the proprietary and closed nature of CDN architectures that do not facilitate external investigation. It is worth to observe that we are not talking about academic or research prototypes, but about systems that represent an important and increasing fraction of the Web business. Indeed, most performance evaluation studies are carried out by CDN companies themselves, and their conclusions are mainly used for marketing purposes.

As our goal is to measure to what extent the use of CDNs improves the user-perceived response time with respect to a non CDN-based service and the dependency of the performance on different network and site parameters, we consider important to investigate at least the following issues, that have been grouped on the basis of the three phases of CDN service.

- *Server selection*: we aim to understand how CDN performance is influenced by external factors such as end-user geographical location, network conditions, time of the day, day of the week.
- *Request routing*: we aim to single out the DNS contribution to the user-perceived performance.
- *Delivery*: we aim to determine what is the number of distinct edge servers used to serve the content of a single Web site, and to what extent this number affects the user-perceived performance.

As regards the investigation of the server selection phase, in order to analyze the client-perceived performance impact of the factors external to the CDN infrastructure, we consider three client locations having different types of network connection, and three measurement periods, during a time of almost two years, characterized by different worldwide network usage. Indeed, previous studies about CDNs [13] have highlighted a considerable variability of the results obtained in experiments carried out during distinct time periods, also due to changes in the CDN network infrastructure. On the other hand, no previous study has analyzed the hourly and daily dependence of measured CDN performance. In our study we do not investigate the CDN ability to select the edge server with the minimum latency to the client as done in [9]. The authors argue that CDN benefits result from avoiding the badly performing servers. However, they do not evaluate the impact of CDNs on user-perceived response time, but analyze only the relative performance of server selection within a given CDN.

In our study we conduct the performance analysis using real pages of Web sites that adopt CDN services rather than using a canonical Web page that reflects the statistical distribution of static objects typically served by a given CDN, as done in [13], or client collected traces, as in [3].

DNS redirection impact on CDN performance has been already investigated in recent studies [3, 12, 13], which confirm that CDNs reduce mean response times, but that DNS-based request routing techniques add a noticeable overhead because of DNS costs (as also observed in [19]). As DNS redirection techniques are transparent to the client, it is difficult to understand their inner mechanism. We measure the DNS resolution time at the client side and analyze to what extent the DNS lookup cost affects the page response time and how this contribution changes in front of variations of external factors such as network conditions.

Furthermore, we avoid a common mistake of other studies that use the mean response time as the most common performance metrics. We claim that the mean is not meaningful in an Internet context, where response times show heavy-tailed distributions. To study how CDN enhancements impact over user-perceived performance, we use the cumulative distribution, the median and 90-percentile of the response time perceived by end users as our main metrics. As observed in [3], the overall page performance is the crucial metric which users are most interested in. Even content providers should focus on it as it directly correlates with the user perception of the quality of service of a CDN system. The page response time corresponds to the interval between the submission of the page request and the arrival at the client of all objects related to the page request. It includes the DNS resolution time, the TCP connection time, all delays at the servers, and the network transmission time.

Another problem that affects a fair performance comparison analysis is that the downloading times can be largely af-

fected by the fact that the Web pages have an extremely different number of embedded objects, each of them has a different file size. Hence, we decided to normalize the CDN page response time with respect to the time when all files are downloaded from the origin server. To this purpose, we retrieve two versions of the same Web page: the former in which objects served by the CDN infrastructure are requested to the edge servers selected by the CDN, the latter in which requests are forced to reach the origin content server(s). Any other comparison parameter would have been arbitrary and much more difficult to tune. This metric, called *speedup*, is defined as the ratio between the non-CDN and the CDN usage case, hence $speedup > 1$ means a performance improvement determined by the usage of CDNs. In particular, we calculate the achievable speedup for the 50-percentile (median) and 90-percentile of the user response time.

4 Evaluation tool

The CDNPerf tool implements the evaluation methodology that has been described in the previous section. This tool consists of three parts, each one related to a specific step of the testing process, that is, experiment configuration, request generation, and output analysis and report.

The main engine of CDNPerf is the browser emulator (also called downloader). This program is an HTTP client that supports most functionalities of the HTTP protocol (both versions 1.0 and 1.1), including persistent connections and pipelining, chuncked encoding transfers, and request redirection. CDNPerf is able to recognize CDN-served embedded objects and hence it can download them from both the customer origin servers and the CDN edge servers. Network failures are handled gracefully with reconnection attempts up to a given limit.

Given a URL referring to a Web page, CDNPerf downloads all its components and records different performance metrics in a log file. After having retrieved the base HTML page, CDNPerf parses it with the goal of identifying each embedded object and determining whether it is being served by a CDN server or not. Then, for each identified server (both origin and edge) CDNPerf downloads each object coming from that server by (re)using a persistent connection, recording also the DNS resolution time as well as the TCP connection setup time. The tool records the total download time for each requested object latency that is, the period intercurring between the request and the first byte of the received response data. In the case of multiple connections (due to network problems or no persistent connection support by the server), multiple connection times are stored into the CDNPerf log file. Finally, when all the objects composing the requested page have been retrieved, the tool records the aggregate response times for the whole page (obtained by summing the total download time for all objects in the page), by distinguishing those obtained by a CDN server and those by the origin server. For resources not served by the CDN, the origin server statistics are used for both aggregate times.

Some characteristics of our browser emulator resemble those provided by the Medusa proxy [3, 12]. Specifically, both tools retrieve two versions of the same Web page, the former served by the CDN infrastructure, the latter served only by the origin server. However, our tool takes into account the impact of DNS resolution time on performance, while the study carried out with Medusa proxy doesn't consider this contribution and also ignore DNS refresh effects due to a fairly small inter-request interval.

Moreover, the Medusa proxy transformation feature is limited to Akamaized URLs [1, 7], while the same feature provided by CDNPerf recognizes also URLs rewritten by other CDN companies (e.g., Speedera) that adopt this routing mechanism.

To simplify the measurement analysis task of the log file of the experiments, we used a structured format that has a syntax similar to YAML [2]. The log is composed by a series of *stanzas*, each one describing one URL download attempt. Each stanza is composed of multiple second level entities (one for each server); the last line contains the aggregate final performance data of downloading. The server entity reports the DNS lookup and connection times as well as a list of Web object entities, each containing latency and total download time for that resource.

The data analyzer component of CDNPerf, implemented through PERL scripts, is responsible for processing the log files and producing the percentiles and cumulative distributions of the selected performance metrics, in particular for the user-perceived response time and DNS lookup time, that are the key metrics of the experimental tests carried out in the following section.

5 Experimental results

From the previous sections it is possible to understand how many network and system parameters can affect the performance of CDN services. These parameters are in part internal to the CDN architecture (e.g., server selection, routing and delivery mechanism, server placement, percentage of embedded objects that are served from edge servers) and in part are out of the CDN provider control (e.g., Internet traffic, bandwidth of the client connection, distance from the origin and edge servers). Most of the previous studies have focused especially on the former aspects. In this paper, we include the internal parameters, but our point of analysis of them starts from the external aspects.

To this purpose, we consider two main Internet traffic conditions (normal traffic, high traffic) and three main qualities for the connection of the client to the CDN architecture (high, medium, low), for a total of six combinations. Due to space

limits, in this section we report the most significant results related to medium and narrow bandwidth for normal traffic, and including also the large bandwidth case for high traffic.

The experiments referring to the normal traffic lasted over two distinct periods: nearly two months from October 5, 2002 to November 30, 2002 and two weeks at the beginning of February 2004, covering a length of time of almost two years. In those periods, we did not observe any special peak of Internet traffic. In fact, we excluded December and periods of special politics and sport events (e.g., international crisis, Olympic games, Wimbledon, Soccer World Cup).

We repeated the same experiments during a period of about one month (from March 14, 2003 to April 10, 2003) in a worldwide Internet condition characterized by heavier traffic, due to international political events (i.e., Iraq crisis). The main visible effect was a round-trip time between the same points clearly higher than that observed in the other periods of observation.

By collecting data over a so long period, we properly analyze how CDN performance change under the effect of the different considered parameters.

In all periods, we considered three types of client-to-CDN connection, that includes the bandwidth of the client connection to Internet, and the distance from the closest CDN edge server. In particular, we have:

High Quality (HQ) location, very large bandwidth of connection to the Internet (16Mbps), 8 network hops to the closest edge server and a round trip time ranging from 1.6 to 12.7 ms with an average of 1.9 ms over 24 hours (the RTT has been measured in a period of high network traffic).

Medium Quality (MQ) location, medium bandwidth of connection to the Internet (4 Mbps), 11 network hops to the closest edge server and a round trip time ranging from 13 to 29 ms with an average of 22 ms.

Low Quality (LQ) location, low bandwidth of connection to the Internet (1 Mbps), 13 network hops to the closest edge server and a round trip time ranging from 26 to 120 ms with an average of 52 ms.

We denote the six possible combinations through Normal-HQ, Normal-MQ, Normal-LQ and High-HQ, High-MQ, High-LQ, for the cases of normal and high traffic, respectively.

For the server side of our tests, we selected 20 Web sites served by two CDN providers (75% and 25% by Akamai [1] and Speedera [20], respectively), that are indicated among their most popular customers. We decided not to report the name of the sites for the sake of privacy and also because no previous authorization has been asked to.

Table 1: Response time and speedup (Normal-MQ)

Content provider	CDN		No CDN		Speed-up	
	median [sec]	90-perc [sec]	median [sec]	90-perc [sec]	median	90-perc
CP_1	2.635	8.252	8.943	16.683	3.393	2.021
CP_2	2.897	12.542	6.195	17.194	2.137	1.370
CP_3	2.747	16.775	6.062	17.693	2.206	1.054
CP_4	5.943	17.021	13.034	17.664	2.193	1.037
CP_5	12.243	35.576	11.671	16.499	0.953	0.463
CP_6	2.697	7.540	12.343	19.130	4.575	2.537
CP_7	1.376	5.974	3.134	9.495	2.277	1.589
CP_8	5.808	11.043	5.753	9.229	0.990	0.8357
CP_9	3.426	11.143	5.402	11.142	1.576	1.000
CP_{10}	1.636	6.974	5.704	10.542	3.485	1.511
CP_{11}	4.951	8.649	6.648	11.501	1.342	1.329
CP_{12}	1.409	8.280	4.767	12.249	3.383	1.479
CP_{13}	4.225	9.279	4.722	8.055	1.117	0.868
CP_{14}	2.390	7.586	2.822	7.797	1.180	1.027
CP_{15}	3.626	7.431	9.716	13.703	2.679	1.843
CP_{16}	1.594	6.184	4.558	8.962	2.859	1.449
CP_{17}	2.017	6.724	3.453	8.407	1.711	1.250
CP_{18}	2.506	9.009	3.307	11.210	1.319	1.244
CP_{19}	3.167	9.693	7.975	13.814	2.517	1.425
CP_{20}	1.618	11.795	3.097	13.266	1.914	1.124

For each Web site, we analyze response times and achieved speedup for all periods of Internet traffic and each geographic location of the client, evaluating also the hourly and daily dependence of measured CDN performance. Then we analyze which is the impact on performance due to the so called internal factors that is, the number of edge servers and the percentage of CDN-served objects. Furthermore, we analyze the impact of the time due to server selection and routing on the total page response time.

5.1 Overall performance of CDN services

The first important test is to verify for which external factors the usage of a CDN service can effectively reduce the response time for Web requests.

Let us first consider the situation of normal traffic. During the two periods referring to the normal traffic we observe consistent measurements for most of Web sites, although almost two years pass between the first and the last experiment. Hence, the first interesting result is that, even if CDN techniques have evolved over this long period of time, the perceived performance in terms of speedup are not changed significantly. This consistency is an important reference point to understand how CDN performance change under different system and network conditions. Table 1 reports in columns 2, 3, 4 and 5 the median and the 90-percentile of the user response time for all Web sites. Columns 6 and 7 report the speedup for the two performance metrics. For three content providers (CP_7, CP_8 and CP_{20}) we report results referred only to the first observation period because afterwards they choose to dismiss partial site distribution with CDN. Table 1 shows the performance for the MQ location, but it is interesting to note that the speedup values are very similar to those achieved from the other HQ and LQ locations.

Hence, we omitted the other two tables, even if it is noteworthy the fact that in case of High network quality (Normal-HQ) the response time is one order of magnitude lower. Table 1 shows that in most of the cases CDN offers a significative performance advantage. In particular, by looking at the 90-percentile of response time, in the 10% of the observed sites the speedup is higher than 2 (this means that the CDN service is more than two times faster than the download from the origin server), in particular up to 2.5 for CP_6. For the 25% of the sites, the speedup is higher than 1.5, and for the 50% of the sites is higher than 1.3. However, in 15% of our observations we found that a CDN service can be slower than that provided by the origin server. In particular, for CP_5 the speedup is limited to 0.46, which means that the CDN response time is two time higher than that from the origin server. At least for this case, we have identified that the possible cause of this performance penalty is due to the high number of edge servers used to deliver objects to the same client.

Let us now consider the situation of high traffic. When the Internet load becomes heavier, the conclusions about CDN performance change significantly.

Table 2 reports the speedup for High-LQ, High-MQ and High-HQ. We can note as the response time tends to increase when network traffic is high. Moreover, by comparing Table 1 and 2, speedup appears to be generally lower. In particular the speedup on 90-percentile is reduced by an increase of pathological cases where CDN cannot deal with congestion. For this reason more than 70% of the content providers in High-LQ have a slowdown in response time when CDN are used by considering 90-percentile, and the highest achieved speedup is 1.76 for CP_{17}. Results from other geographic locations share many common characteristics. The results of the various clients are similar for median values (even if the better connectivity in High-HQ offers greater speedup), while, due to the less predictable nature of congested networks, the results of 90-percentile show some differences, with only 18% of content providers with a speedup lesser than 1 for High-MQ and nearly 40% for the case High-HQ. The speedup is however still generally reduced with respect to the case of normal network load. From the comparison with the situation of normal traffic, we note that few content providers increase their performance. In many cases this can be explained with changes in the CDN usage policy: for example for CP_5 the performance is increased as the usage of edge servers is modified (as we can see in greater detail in section 5.3). Moreover three content providers (CP_7, CP_8 at least for home page and CP_{20}) choose to dismiss partial site distribution with CDN (hence those providers are missing in tables related to experiment 2). There are, however, many sites (such as CP_1) where CDN usage (and to a limited extent, also performance) is comparable in both experiments.

Another advantage of CDN usage verified in our experiments is their ability to reduce variance in response time. Figure 2 shows the cumulative probability distribution of user response time for a sample site (CP_1) in both network conditions (in both cases we used data related to the location MQ). We choose this content provider as an example because it is the most popular site we studied, but the considerations can be applied to most of the sites. Not withstanding the performance differences, the figure shows that the cumulative probability curve of CDN is far steeper: this means that CDN can be effective in reducing performance variability.

Table 2: Speedup (High-LQ, High-MQ, High-HQ)

Content provider	High-LQ		High-MQ		High-HQ	
	median	90-perc	median	90-perc	median	90-perc
CP_1	2.652	0.826	2.678	2.201	5.518	4.825
CP_2	1.613	0.991	1.495	1.106	1.436	0.976
CP_3	0.985	1.064	0.694	0.752	n/a	n/a
CP_4	1.183	0.667	1.563	1.743	1.268	0.757
CP_5	1.031	0.770	0.845	0.867	0.930	0.729
CP_6	1.223	0.780	1.269	1.000	1.461	1.268
CP_9	1.509	1.266	1.317	1.746	1.083	0.927
CP_{10}	1.563	0.582	1.546	1.170	2.610	2.225
CP_{11}	1.812	1.023	1.679	1.221	2.767	1.264
CP_{12}	2.042	0.781	2.154	1.822	1.480	1.313
CP_{13}	1.034	0.576	1.054	1.245	0.855	0.870
CP_{15}	1.836	0.717	2.003	1.534	2.275	1.742
CP_{16}	1.422	0.581	1.734	1.291	2.693	1.592
CP_{17}	2.510	1.759	2.284	1.561	3.696	3.497
CP_{18}	1.116	0.850	1.026	0.993	1.554	1.269
CP_{19}	1.649	0.750	1.421	1.255	2.768	2.047

By comparing Figure 2(a) and 2(b) we can also have a visual confirmation of the Internet congestion occurring at that time: Figure 2(b) shows smoother curves, that tend to have a higher probability of pathological cases with a very high response time. This also tends to reduce the advantage of CDN on 90-percentile, up to the case where CDN services do not achieve any further performance improvement. From this figure we can conclude that in the case of high network traffic, CDN can still offer some performance gain, but this advantage is much less evident with respect to the case of low-medium Internet traffic. Even the appreciable CDN ability to limit performance variance is reduced as well.

5.2 Hourly dependence of CDN performance

The hour of the day and the day of the week are other external factors that influence CDN performance and that have not been previously examined. Quite surprisingly, we found that most Web sites show a similar behavior, hence we report only two graphs (Figure 3) that show the median response times as a function of the hour of the day for normal and high traffic condition, respectively. Another premise is in order. There is a strict correlation between night hours and weekend days performance, and between daylight hours and week days performance. Hence, although we focus just on the day hours, other conclusions can be easily obtained.

In the case of normal traffic, Figure 3(a) shows that CDN

performance are clearly better than the no-CDN case: the median values are about one third of the no-CDN case, and the CDN curve has less and smaller spikes than the no-CDN case. This confirms the previous observation that CDN services can effectively reduce the variance of the response time when the network load is limited (as shown also in Figure 2(a)). In Figure 3(a) we consider Normal-MQ, but same conclusions hold for the other client locations. We also note that edge servers belonging to the same physical region tend to show similar access patterns with respect to the hour of the day. This is the reason for the slight increase of the response time during day hours with respect to night hours. On the other hand, the origin server which receives request from different geographic locations shows less predictable and regular response times.

In the case of high network traffic (Figure 3(b)), the client location has a significant impact on performance. When connections have low-medium quality (High-LQ and High-MQ), CDNs are no longer able to compensate possible congestion due to the network links. Hence, their response time hourly graph shows a great difference between daytime and night. Moreover, CDNs seem unable to limit response time nor its variance, thus showing a behavior comparable with the no-CDN case. However, it is worth to note that heavy network load does not automatically result in bad performance. When the quality of the client connection is good (High-HQ), the performance results of CDNs are very similar to those shown in Figure 3(a), thus not showing any trace of large variance even in the case of high traffic.

5.3 Effects of internal mechanisms

We now pass to consider the impact on CDN performance due to some internal factors. We consider the percentage of objects served by the edge servers and the number of edge servers used to deliver content to the same client. Table 3 shows how many embedded objects contains the home page of each Web site (column 2), and which absolute number and percentage are served from CDN edge servers (column 3 and 4, respectively). The last column reports the number of edge servers used to deliver the respective embedded objects.

Table 3: Internal factors

Content provider	Embedded objects			Edge servers
	Total	CDN-served	%	
CP_1	31	29	90%	3
CP_2	58	56	97%	1
CP_3	44	43	98%	1
CP_4	44	42	95%	1
CP_5	29	13	45%	13
CP_6	33	29	88%	1
CP_7	9	9	100%	6
CP_8	21	20	95%	1
CP_9	18	11	61%	1
CP_{10}	29	29	100%	2
CP_{11}	26	13	50%	1
CP_{12}	13	13	100%	1
CP_{13}	19	19	100%	1
CP_{14}	29	23	79%	1
CP_{15}	29	29	100%	1
CP_{16}	25	25	100%	1
CP_{17}	10	10	100%	1
CP_{18}	26	19	73%	1
CP_{19}	31	29	94%	1
CP_{20}	14	14	100%	1

The most interesting aspect emerging from Table 3 is that the large majority of CDNs use only one edge server to deliver all embedded objects to the same client. The use of multiple edge servers for the content providers CP_1 and CP_{10} is related to the logical subdivisions of the site content delivery. These sites use a main edge server for most embedded objects, and other edge servers for specific functions, such as dynamically generated images and/or advertising banners.

The most significative exception is the content provider CP_5, that uses a different edge server for each CDN-served embedded object. This is an interesting representative case that deserves some further discussions. The first observation is that this site performs poorly and this is due to different effects. If each edge server has to start a new TCP connection for each embedded object, the positive effects of HTTP/1.1 persistent connections cannot be exploited. Moreover, as each edge server belongs to the same network area, the possible benefits of parallel download are reduced. The poor performance of CP_5, together with the choice of the large majority of CDN providers to use only one edge server to deliver content to the same client, confirm the results found by Kangasharju et al. [10], showing that the retrieval of objects on the same Web page from multiple servers may cause a performance degradation. It is interesting to observe that the performance of the content provider CP_5 shows great improvements in the period of high Internet traffic, when its internal architecture was changed to a single server delivery basis.

The relationship between speedup and percentage of CDN-served objects is shown in Figure 4. We reported the speedup vs. percentage of CDN-served objects for both median (Figure 4(a)) and 90-percentile (Figure 4(b)). Both graphs show that: (1) to maximize the CDN performance benefits, they must be heavily used; (2) heavy usage of CDNs is only a necessary condition, and it is not sufficient to guarantee high speedup. From the first observation, we can conclude that CDNs are a powerful tool to increase Web performance. This confirms the results in [12], even if the analysis conducted through the Medusa Proxy tool tends to underestimate the impact of DNS lookup time on performance.

CDNs must be used carefully in order to maximize their potential benefits. From Figure 4 we can see that the sites relaying completely on the CDN infrastructure for the delivery of embedded objects show large speedup differences. However, it is interesting to see that the best performing sites seem

to be those with lower percentages of CDN-served objects. This apparently counter-intuitive result can be motivated by the fact that, when some objects are served by the origin server, there is a load distribution between the edge server (more powerful and with better connection) and the origin server itself (which has already an open connection with the client) that may have enough spare resources to use. This system, however, requires a careful tuning in order to avoid congestion at the origin server.

The same relationship between the fraction of CDN-served embedded objects and performance has been observed under different network conditions and from different client locations. Hence, we can conclude that our observations have a general validity because they are neither related to geographic location nor to the network traffic conditions.

5.4 Impact of the routing mechanism

DNS-based redirection is the main mechanism for request routing in the considered CDNs, as explained in Section 2. In [13] Krishnamurthy et al. found that, when sophisticated DNS servers are used, the name resolution time tends to increase with respect to traditional DNS systems. Our experiments confirm this result, even if there are some few remarkable exceptions. The results are shown in Figure 5, which presents the cumulative distribution of the DNS lookup time. In particular, Figure 5(b) shows clearly that the DNS lookup time tends to be more expensive when CDNs are used.

From some client locations we find hints of the presence of a multi-tier DNS lookup process: for more than 50% of the content providers, the cumulative distribution presents one or more steps, as shown in Figure 5(a). When complex DNS caching and resolution mechanisms are used for both the CDN and the Web site, it is more difficult to identify a clear performance difference on the DNS response time observed in the case of CDN usage.

It is also interesting to compare the DNS lookup time in the case of normal and heavy network traffic. Figure 6 shows the cumulative distribution of the median contribution of DNS lookup time to the total response time over the various content providers. In the case of normal network load, for any client location (Figure 6(a) refers to Normal-MQ, but the results are very similar for Normal-HQ and Normal-LQ), we observe that the DNS lookup time can be up to nearly 40% of the global response time in the case of CDN usage, while it is no more than 20% when CDNs are not used. Moreover, the DNS lookup time contributes for more than a tenth of the global service time in 70% of the cases in which a CDN is used. On the other hand, when CDNs are not used, the DNS lookup contributes for more than a tenth of the global service time only in less than 40% of the content providers. Therefore, we can conclude that DNS resolution can be a significative part of the service time provided by CDNs. Indeed, this greater influence of the DNS lookup time when CDN are used is the sum of two combined effects: the DNS lookup time tends to be increased by the use of CDN; the global download time is reduced by the use of CDN in the case of normal traffic.

When the traffic is higher, the network impact tends to increase the download time much more than the DNS lookup time. As shown in Figure 6(b), the consequence is that the DNS lookup time has almost a negligible impact on the total download time for high traffic conditions.

6 Conclusions

This paper addresses the problem of evaluating user perceived CDN performance gain: we developed a powerful and flexible tool for measuring CDN performance by emulating a Web browser and collecting many data on the various part of the client request service. The output of our tool can be then processed with different log processors in order to extract information from the data.

Our experiments focus on real sites, considering then the performance benefit in downloading a whole page, with all embedded objects instead of focusing only on the single Web objects. Moreover we evaluate also the DNS lookup time, in order to validate the hypothesis that in CDN, when DNS-based redirection schemes are typically used, the DNS resolution time can became a significative part of the response time. We study different CDN companies using partial site delivery and the experiments performed over a long period of time cover a wide range of situations both on the side of network traffic and of geographic locations, with different network resources available to clients. In our experiments we also evaluate the performance benefit of CDN with respect to different factors, such as the percentage of embedded object actually served by the CDN, the number of edge servers used as well as the hour of the day and the day of the week.

Our observations shows that indeed CDN can offer significative performance gain over the traditional solution with a centralized, and possibly far, Web server. Moreover CDNs reduce variance in response time because typically less hops (and hence less points where delay can occur) are required to reach the edge server. However we found that under heavy network load, when congestion arise, the benefit of using CDNs is reduced; in particular when considering the 90-percentile of the response time it is possible that the CDN is as fast as or even slower than the situation with only one site. Moreover in case of congestion the CDN can show heavy time-dependent behavior, with response times far higher during the busiest hours of the day.

From the comparison of site structures and performance, we found that there is a strong correlation between the achieved speedup and the fraction of the site CDN-served. However, a heavy use of CDN for content distributions is a requisite but is not sufficient to achieve high speedup, as our

experiments pointed out.

We also validated two results presented in other articles: we found that CDN achieve better performance when only few (typically one) edge servers are used instead of many servers (up to one for every object), validating a simulative result; moreover we confirmed the observation that, in CDNs, the DNS resolution time can be a significative part of the total response time, even if we found that when congestion occurs this result can be no more true because download time can grow more than DNS lookup time.

The summary of our experiments is that CDN are a powerful mechanism that can contribute in increasing the user-perceived performance of the Web. However they are not a *panacea* that allows to arbitrarily improve the performance of a Web site: careful site and content distribution design is required to fully exploit the power of CDN. Moreover in case of critical network condition, when congestion occurs, the performance of CDN can be reduced as well as for every node in a congested network.

References

[1] Akamai Tech. `http://www.akamai.com/`.

[2] O. Ben-Kiki, C. Evans, and B. Ingerson. Yaml ain't markup language, Mar. 2003. `http://www.yaml.org/`.

[3] L. Bent and G. Voelker. Whole page performance. In *Proc. of 7th Int'l Web Caching Workshop and Content Delivery Workshop*, Boulder, CO, Aug. 2002.

[4] B. Cain, A. Barbir, N. R., and O. Spatscheck. *Known CDN request routing mechanisms*. Network Working Group, Internet-Draft, Nov. 2002.

[5] V. Cardellini, E. Casalicchio, M. Colajanni, and P. S. Yu. The state of the art in locally distributed Web-server systems. *ACM Computing Surveys*, 34(2):263–311, June 2002.

[6] V. Cardellini, M. Colajanni, and P. S. Yu. Request redirection algorithms for distributed Web systems. *IEEE Trans. on Parallel and Distributed Systems*, 14(5), May 2003.

[7] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. *IEEE Internet Computing*, 6(5):50–58, Sept./Oct. 2002.

[8] Edge Side Includes. `http://www.esi.org/`.

[9] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek. The measured performance of Content Distribution Networks. *Computer Commun.*, 24(1-2):202–206, Feb. 2001.

[10] J. Kangasharju, K. W. Ross, and J. W. Roberts. Performance evaluation of redirection schemes in Content Distribution Networks. *Computer Commun.*, 24(1-2):207–214, Feb. 2001.

[11] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi. Web caching with consistent hashing. *Computer Networks*, 31(11-16):1203–1213, Feb. 1999.

[12] M. Koletsou and G. Voelker. The Medusa proxy: A tool for exploring user-perceived Web performance. In *Proc. of 6th Int'l Web Caching Workshop and Content Distribution Workshop*, June 2001.

[13] B. Krishnamurthy, C. E. Wills, and Y. Zhang. On the use and performance of Content Distribution Networks. In *Proc. of SIGCOMM IMW 2001*, pages 169–182, Nov. 2001.

[14] P. Kulkarni, W. Gong, and P. Shenoy. Scalable techniques for memory-efficient CDN simulations. In *Proc. of 12th World Wide Web Conf. (WWW2003)*, Budapest, Hungary, May 2003.

[15] Z. M. Mao, C. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A precise and efficient evaluation of the proximity between Web clients and their local DNS servers. In *Proc. of USENIX Ann. Tech. Conf.*, June 2002.

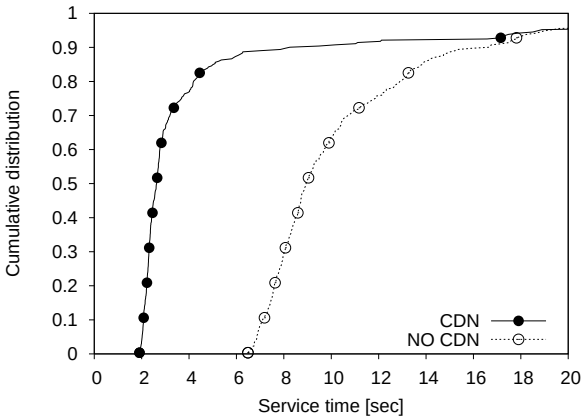
[16] Mirror Image Internet. `http://www.mirror-image.com/`.

[17] M. Rabinovich and O. Spatscheck. *Web Caching and Replication*. Addison Wesley, 2002.

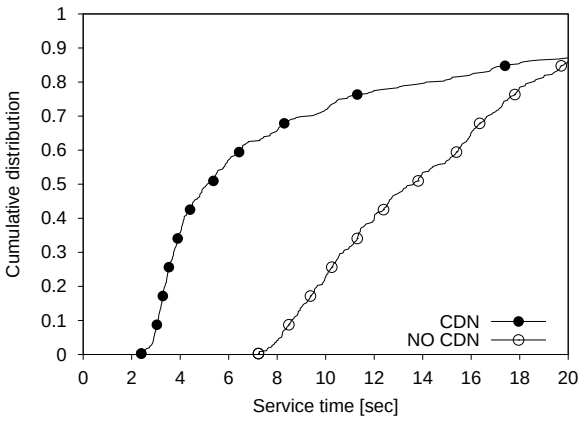
[18] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and L. H. M. An analysis of Internet content delivery systems. In *Proc. of 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, Dec. 2002.

[19] A. Shaikh, R. Tewari, and M. Agrawal. On the effectiveness of DNS-based server selection. In *Proc. of IEEE Infocom 2001*, pages 1801–1810, Apr. 2001.

[20] Speedera Networks. `http://www.speedera.com/`.

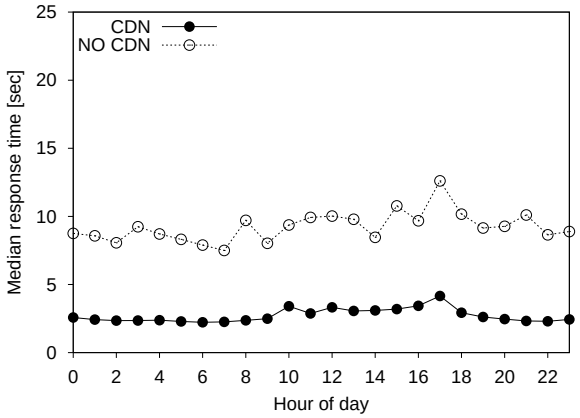


(a) Normal traffic

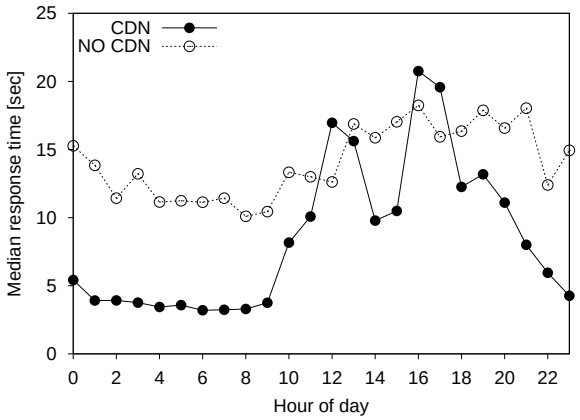


(b) High traffic

Figure 2: Cumulative probability of the user response time for CP_1

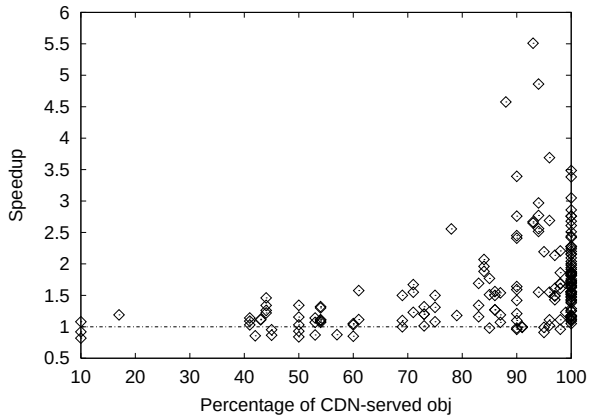


(a) Normal-MQ (similar for High-HQ)

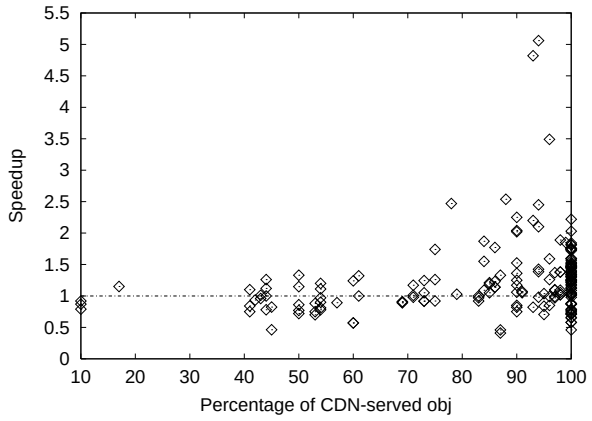


(b) High-MQ, High-LQ

Figure 3: Hourly dependence on performance.

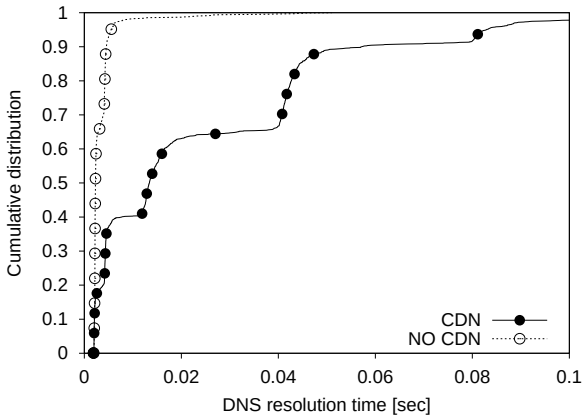


(a) median

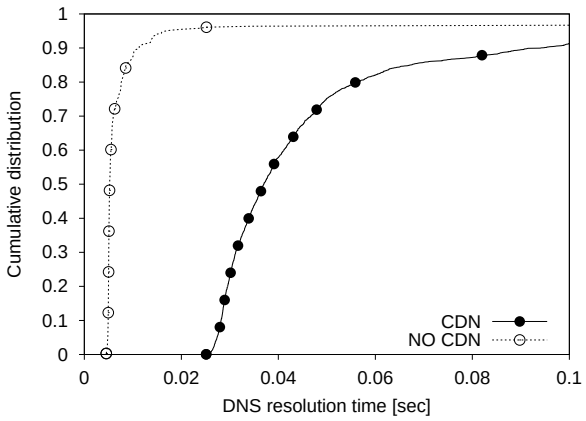


(b) 90-percentile

Figure 4: Relationship between speedup and percentage of CDN-served objects.

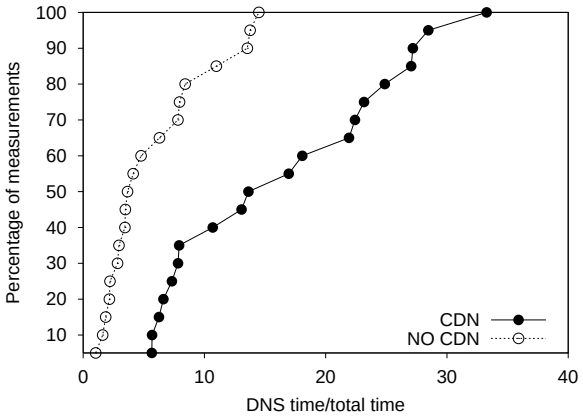


(a) Multi-tier DNS lookup

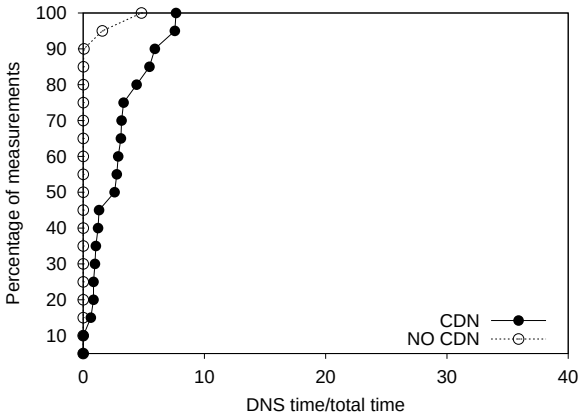


(b) One-tier DNS lookup

Figure 5: Cumulative probability of DNS lookup time (normal traffic).



(a) Normal traffic



(b) High traffic

Figure 6: Cumulative probability of DNS lookup time.