On the Use and Performance of Content Distribution Networks

Balachander Krishnamurthy, Craig Wills, Yin Zhang

Abstract— **Content distribution networks (CDNs) are a mechanism to deliver content to end users on behalf of origin Web sites. Content distribution offloads work from origin servers by serving some or all of the contents of Web pages. We found an order of magnitude increase in the number and percentage of popular origin sites using CDNs between November 1999 and December 2000.**

In this paper we discuss how CDNs are commonly used on the Web and define a methodology to study how well they perform. A performance study was conducted over a period of months on a set of CDN companies employing the techniques of DNS redirection and URL rewriting to balance load among their servers. Some CDNs generally provide better results than others when we examine results from a set of clients. The performance of one CDN company clearly improved between the two testing periods in our study due to a dramatic increase in the number of distinct servers employed in its network. More generally, the results indicate that use of a DNS lookup in the critical path of a resource retrieval does not generally result in better server choices being made relative to client response time in either average or worst case situations.

I. Introduction

A few thousand Web sites receive a significant fraction of request traffic. HTTP protocol changes [1] have enabled access latency reduction via improved caching, longer-lived HTTP connections, and the ability to download selective portions of a resource. Caching aims to move content closer to users to help diminish load on origin servers, eliminate redundant data traversal on the network, and reduce user-perceived latency. Traditional caching has limited effectiveness due to diversity of resource access, increasing dynamic content, and concerns about consistency of cached responses. Busy sites, however, need additional mechanisms to deliver acceptable performance.

A *Content Distribution Network* (CDN) consists of a *collection* of (non-origin) servers that attempt to offload work from origin servers by delivering content on their behalf. The servers belonging to a CDN may be located at the same site as the origin server, or at different locations around the network, with some or all of the origin server's content cached or replicated amongst the CDN servers. For each request, the CDN attempts to locate a CDN server close to the client to serve the request, where the notion of "close" could include geographical, topological, or latency considerations. With content distribution, the origin servers have control over the content and can make separate arrangements with servers that distribute content on their behalf.

Balachander Krishnamurthy and Yin Zhang are with AT&T Labs–Research, Florham Park, NJ, Craig Wills is with Worcester Polytechnic Institute, Worcester, MA. e-mail:bala@research.att.com, cew@cs.wpi.edu, yzhang@research.att.com Contact author: Balachander Krishnamurthy, fax: 973-360-8077, 180 Park Avenue, Florham Park, NJ 07932

While CDNs have been created by a number of companies and these CDNs are being used to serve content on behalf of popular origin server sites, there is little that has been published on the extent to which CDNs are being used and their relative performance in serving content. This work seeks to address this void by gathering use and performance data to answer a number of research questions:

1. What CDN techniques are being employed and how does the choice of these techniques influence performance?
2. What is the extent to which CDNs are being used by popular origin server sites?
3. What is the nature of content being offloaded by origin servers to CDNs?
4. What methodology can be used to measure the relative performance of CDNs given that operational details of CDNs are not public?
5. How are specific CDNs performing in serving content both relative to origin servers and among themselves?
6. What conclusions can be drawn about the operations of CDNs and their effect on client-perceived performance?

The paper describes the answers we have found to these research questions through a large-scale, client-centric study that gathered data over the time period of September 2000 to January 2001. The data are from proxy logs and content retrieved from origin servers and CDNs. The data are primarily concerned with static image content, but also include preliminary data on CDNs being used to serve streaming media content.

The rest of the paper is organized as follows: Section II provides background on the studied CDN techniques. Section III examines how CDNs are being used today. Sec-

TABLE I
REFERENCED CONTENT DISTRIBUTION NETWORKS AND THEIR URLS

CDN	URL	CDN	URL
Adero	www.adero.com	Mirror Image	www.mirror-image.com
Akamai	www.akamai.com	Navisite	www.navisite.com
Clearway	www.clearway.com	NetCaching	www.netcaching.com
Digital Island	www.digitalisland.com	Solidspeed	www.solidspeed.com
Exodus	www.exodus.com	Speedera	www.speedera.com
Fasttide	www.fasttide.com	IntelliDNS	www.unitechnetworks.com
Intel	www.intelonline.com	Yahoo!Broadcast	business.broadcast.com

tion IV describes the methodology of a performance study of CDNs with Section V detailing the results of this study. Pointing to future work, Section VI presents preliminary results on the use and performance of CDNs to serve streaming content. Section VII summarizes the work and highlights what has been learned from it.

II. BACKGROUND ON CDN TECHNIQUES

The first step taken by a client to retrieve the content for a URL is to resolve the server name portion of the URL to the IP address of a machine containing the URL content. This resolution is done with a Domain Name System (DNS) lookup by the client. The resolution causes a DNS request to be sent to a local DNS server. If the local DNS server does not have the address mapping already in its cache, the local DNS server sends a query to the authoritative DNS server for the given server name.

Servers in a content distribution network (CDN) are located at different locations in the Internet. A primary issue for a CDN is how to direct client requests for an object served by the CDN to a particular server within the network. DNS redirection and URL rewriting are two of the commonly used techniques for directing client requests to a particular server in a distributed network of content servers.

For the DNS redirection technique, the authoritative DNS name server is controlled by the CDN. The technique is termed DNS redirection because when this authoritative DNS server receives the DNS request from the client (actually from the client's local DNS server [2]) the DNS server redirects the request by resolving the CDN server name to the IP address of one content server. This resolution is done based on factors such as the availability of resources and network conditions. When the authoritative DNS server replies with the IP address mapping it also includes a time-to-live for the mapping. Generally the reply has a low TTL so that the CDN can change the mapping quickly to facilitate load balancing among its servers.

There are two types of CDNs using the DNS redirection technique: full- and partial-site content delivery. With full-site content delivery, the origin server is largely hidden except to the CDN; the origin site modifies its DNS zone file (a zone is a subtree of the DNS hierarchy that is separately administered) to reflect the authoritative DNS server provided by the CDN company. Adero, NetCaching, and Unitech Networks' IntelliDNS are examples of CDNs delivering the full content for origin sites (see Table I for the set of CDNs that we came across in our study.). All requests for the origin server are directed, via DNS, to a CDN server. The CDN server either serves the content from its cache or forwards on the request to origin server.

With partial-site content delivery, the origin site modifies the embedded URLs for objects (primarily images) to be served by the CDN so that the host names in the URLs are resolved by the CDN's DNS server. The actual syntax of the rewritten URL varies with the CDN. For example, the Speedera CDN changes `www.foo.com/bar.gif` to `foo.speedera.net/www.foo.com/bar.gif`, The host name in the modified URL can be in the same domain as the origin site or in a different domain. In the former case, besides modifying embedded URLs, the origin site also needs to modify its zone file. Akamai, Digital Island, Mirrorimage, Solidspeed, and Speedera are examples of CDN companies delivering partial content.

The other content distribution technique used by CDNs in our study is URL rewriting whereby an origin server rewrites URL links as part of dynamically generating pages to redirect clients to different content servers. The Clearway CDN company, for example, identifies objects on customer origin sites that are likely to gain from replication and pushes them to CDN mirror servers. At resource access time, the page is dynamically rewritten with the IP address of one of the mirror servers, thus avoiding the need for a DNS lookup, so that the client can directly retrieve the replicated objects.

Fasttide is an example of a company that combines URL rewriting with DNS redirection. Fasttide uses URL rewriting to identify a particular Fasttide server which might resolve to the IP address of another CDN server when that Fasttide server name is resolved.

III. USE OF CONTENT DISTRIBUTION NETWORKS

The first part of our study examined how CDNs are being used to serve content in the Web and the nature of the content served. In [3] it was reported that only 1-2% of approximately 670 popular Web sites were employing CDNs to serve content based on data gathered in November, 1999. As a follow-up to this study we compiled two lists of popular sites for determining the use of CDNs by popular Web sites. The first list, "HOTMM127," contained 127 sites obtained by obtaining the Media Metrix top 50 list [4] and the 100hot.com list [5]. The second list, "URL588-MM500," was larger, containing 1030 sites obtained by combining the list of servers used in [6] and the Media Metrix top 500 list. Home pages from sites on each of these two lists were retrieved on a daily basis during November and December 2000 for 60 days.

In analyzing the home pages and their embedded images we found that 39 (31%) of the HOTMM127 sites and 177 (17%) of the URL588-MM500 sites used a CDN to serve some of the content on the page. These results indicate a clear increase in the number and percentage of popular origin sites using CDNs to serve content in comparison with the results in [3]. CDN-served content was identified by the presence of a CDN provider name in the server portion of a URL. We also used the output of the *dig* (Domain Information Groper) utility, which does a DNS lookup, to look for a CDN provider as the authoritative name server for other server names we encountered. Of the 39 HOTMM127 sites using CDNs, 37 used Akamai and two used Digital Island. Of the 177 URL588-MM500 sites using CDNs, 165 used Akamai, 20 used Digital Island and one used Adero. Some of these sites used more than one CDN.

We also wanted to examine the content served by CDNs that were *not* well-represented in the list of popular sites. We therefore created a list of 58 Web sites ("CDN58") believed to be served by a CDN provider other than Akamai and gathered data for a few weeks in December, 2000 using the same methodology. This list included 10 sites using Adero, 13 using Digital Island, 11 using Solidspeed and 24 using Speedera.

A. Change Characteristics of CDN-Served Content

To better understand the characteristics of CDN-served content we used the results of our periodic crawl of Web sites over a 60 day period to examine the rate of change of the content. We analyzed the change characteristics of this content from two perspectives: 1) how frequently the set of URLs served by a CDN change; and 2) how frequently the same URL served by a CDN changes. The results of this analysis are shown in Table II.

The results show that the set of URLs served by CDNs changes little for each of the three sets of home pages. 86-94% of CDN-served objects were seen in a previously retrieved version of the containing home page. The content of a CDN-served URL changes little—less than one percent based on changes in the MD5 checksum. This result is not surprising given that images constitute almost all of the CDN-served content. The change frequency is a bit higher when considering cases of an HTTP `no-cache` directive (used to bypass the cache and fetch resource directly from the origin server, 0-2%) or a missing or changed lmodtime (last modification time, 2%). These results indicate that these CDNs are serving little, if any, dynamically generated content that is actually changing on each access. In a small number of cases, we found a new URL with the same contents (based on the same MD5 checksum) as a previously seen URL.

B. Nature of HTTP-Requested CDN Content

The results from our periodic crawl of Web sites provides one perspective on the CDN-served content at these sites. However, they do not directly measure the nature of CDN-served content that has been served based on user HTTP requests. To analyze the CDN-served content served due to user requests, we extracted data from two large proxy log sets—the proxy log traces from nine NLANR sites recorded over the course of a week in January 2001 [7] and the traces from three sites of a large manufacturing company recorded over the course of a week in September 2000. The NLANR traces consist of 33 million accesses from 5023 client IP addresses with the company traces consisting of 114 million accesses from 155,000 client IP addresses. These proxy logs were chosen because they are timely and represent two large and distinct user groups. Although not necessarily representative of all user groups, we can summarize the sets of logs as follows:

- Images account for 96-98% of the CDN-served objects, but only 40-60% of the CDN-served bytes. The unaccounted for objects in these proxy logs account for a small percentage of objects, but a substantial percentage of bytes. Objects were classified based on content type and URL suffix type. The results simply show that some large objects could not be clearly classified as either images or streaming content using this method. For example, some of these objects are unclassified "application/octet-stream"

TABLE II
DAILY CHANGE CHARACTERISTICS OF CDN-SERVED OBJECTS FROM HOME PAGES OF GIVEN SITES

	HOTMM127	URL588-MM500	CDN58
Objects (1000s)	24.9	75.0	15.6
Previously Seen URL (%)	89	86	94
Previously Seen URL with No-Cache Directive (%)	0	1	2
Previously Seen URL with LModTime Changed or Missing (%)	2	2	2
Previously Seen URL with MD5 Changed (%)	0.2	0.2	0.3
New URL, but Previously Seen MD5 (%)	1	4	0

data.

- Among the CDNs, Akamai serves over 85-98% of the CDN-served objects in the proxy site logs and a comparable range of the CDN-served bytes.
- Focusing on images, which predominate the CDN-served object requests, the logged cache hit rates of CDN-served images ranges from 30-80% and cache hit rates of 25-60% for non-CDN-served images. Cache hit rates are generally 20-30% higher for CDN-served content when comparing the two hit rates from the same proxy site. These results indicate some correlation between frequently requested and CDN-served image content.

IV. PERFORMANCE STUDY OF CONTENT DISTRIBUTION NETWORKS

The second aspect of our study of CDNs examined their performance in serving images, the predominant content type served by CDNs. The performance of CDNs can be measured in many ways—how many requests are offloaded from origin servers, their impact on client-perceived latency and their ability to efficiently load balance requests amongst a set of CDN servers. Access to CDN log data is needed to measure the actual number of requests offloaded by CDN servers, but the other two performance indicators can be measured through an active measurement study.

The study focuses on the client-perceived performance of CDNs using DNS redirection and URL rewriting. Little work has been done on measuring the performance of CDNs. One piece of work briefly examined how content distribution servers improved latency when compared to throughput from the origin servers [3]. Johnson et al. [8] assessed the degree to which two different CDNs optimally redirected requests among their mirrors. By studying three clients downloading a single 3-4KB image they found that the CDNs appeared to use the DNS mechanisms not necessarily to select optimal servers, but to avoid selection of bad servers, though it is hard to know how to generalize their study given its limited scope. In more recent work, the mirroring proxy Medusa [9] examines a single CDN (Akamai) in a technique similar to [3]. The study presents performance improvement results from a single user point of view for a small workload. DNS timeout effects are ignored in their study due to a fairly small inter-request interval.

Our study evaluates response time performance of CDNs in delivering content to a set of client sites. Because the study is based on client-side measurements, it can be used to better understand performance issues for CDNs using techniques visible at the client—DNS redirection and URL rewriting. The study could be used by content providers seeking to evaluate potential performance improvements by contracting with a CDN; the content provider could perform the study from a cross-section of their own customer sites to better understand which of the CDNs will provide better response time relative to servers at the content provider site.

This performance study concentrates on the delivery of image content to Web clients. The primary performance measure used for the assessment is the client-perceived response latency for locating a specific content distribution server using DNS and then downloading a set of images from the CDN server.

The performance study is appropriate for a number of reasons. First, as shown in Section III, the distribution of static content in the form of images is a common feature shared by many CDNs. Second, a primary purpose of CDNs is to move content closer to end users, thereby reducing the latencies for users to retrieve the content. Third, our methodology tests both the additional delay and the effectiveness of CDNs using DNS to direct requests away from loaded servers. Fourth, our methodology can be applied to CDNs without bias. Finally, the methodology can be applied to origin sites to create a baseline to assess the relative performance of CDNs.

We begin by outlining our methodology starting with the construction of a canonical page used in the study. We then describe our experiment, including the measurement infrastructure, CDNs, origin sites and clients used in the study.

A. Content for Study

We began this part of our study by determining realistic distributions for the number and sizes of embedded images expected on a Web page. Our motivation was to construct a "canonical page" that reflects these distributions for static images as typically served by CDNs. For each CDN we then construct a list of image URLs currently served by the CDN from a single origin server, that closely match in size those on the canonical page. By doing so, we can draw from different CDNs a set of items similar to those we retrieve from other CDNs, and likewise similar to those we would find a CDN serving if it were used to serve content for Web pages of the origin server.

We first gathered image data from the home pages and their immediate descendents of the top popular Web sites as identified by MediaMetrix [4] and 100hot.com [5]. We also gathered data on images known to be served by the different CDNs; these formed the pool from which we selected the set of images to construct each CDN's canonical page. Table III shows the resulting median and mean image sizes when we originally gathered the data in September 2000 and when we gathered the data again in January 2001. The four CDNs shown are those for which we gathered data in both time periods.

The distribution of the image sizes yields approximate log-normal distributions [10, Section 10.4] for all sets. In addition, the size distribution for all embedded images is similar to the size distribution for images served by CDNs.

Similarly, we studied the number of embedded images on these Web pages, with median and mean results also shown in Table III for both periods of data collection. Pages with content served by Speedera show smaller median values for the number of embedded images, with a fair amount of consistency for the other sets of pages. In developing a canonical page of images for our study we decided on a page with 18 embedded images using the empirical distribution of sizes for embedded images from the popular Web sites to randomly generate the size distribution of these images.

Because the percentage of embedded images on a page being served by a CDN varies, we also examine variations on our canonical page of 18 images. To do so we have drawn a set of 54 image sizes from the distribution for testing the downloading of a large number of images. Some pages contain fewer than 18 images or images may be cached as indicated by results from Section III. Therefore in our tests we actually download 54 images, but record the intermediate measurements for downloading 6, 12 and 18 images. All results reported in this paper are for downloading the first 18 images unless otherwise specified. The sizes of all 54 images listed in the order they are retrieved in our tests are: 49, 1836, 54, 2291, 1272, 6635, 78, 6840, 117, 2175, 912, 462, 12902, 2182, 36, 35, 2209, 307 (end of canonical page), 9776, 3020, 384, 2425, 354, 430, 788, 5732, 93, 12384, 160, 417 571, 85, 3526, 641, 3451, 334, 61, 11824, 9753, 3541, 1428, 880, 82, 9429, 124, 1118, 2282, 115, 91, 59, 3927, 12705, 46 and 291 bytes.

B. Content Distribution Networks and Origin Sites for Study

We began by creating an instance of the canonical page using images served by that CDN. To find images served by the CDN, we used the results from the background study described in Section IV-A. Table IV shows the source of the images for each of the six CDNs we evaluated in January 2001. Using this source of images, we found the image closest in size to each of the 54 images in our complete set of image sizes. The table shows the average size difference in bytes between the images served by the CDN and those image sizes on the list. The relatively small averages indicate success in matching actual images with target image sizes.

As a means for comparison, we also created instances of the canonical page for popular U.S. and international origin sites from images being served by those sites. In comparing CDN performance with that of origin servers a seemingly ideal test would retrieve two versions of an origin server's page—one using a CDN to serve content and the other not. This approach was used for two previous studies [3], [9]. However, a straightforward application of this test for a origin server employing a CDN for partial-site DNS redirection causes content to be retrieved from an origin server that it expects to be retrieved from the CDN. Even if the origin server contains the content it is unlikely to be optimized for quick retrieval so comparisons to the retrieval time from the CDN are suspect.

We thus choose not to use this approach because it only works with partial-site DNS redirection and even then the origin server may not be optimized to serve such content to clients. We instead choose to use a methodology that could be used with other CDN selection techniques and to apply it to popular origin servers that would be expecting to serve the requested content.

C. Study Description

The algorithm used in the study for a client to retrieve a set of images from a server mimics the steps taken by a user agent. The steps taken by a client are as follows.

TABLE III
STATISTICS FOR THE SIZE AND NUMBER OF EMBEDDED IMAGES ON GIVEN WEB PAGES

	Sept. 2000					Jan. 2001				
	Image Byte Size		Images on Page		No.	Image Byte Size		Images on Page		No.
Source of Pages	Med.	Mean	Med.	Mean	Pages	Med.	Mean	Med.	Mean	Pages
Hot100 Sites	808	2231	17	19.3	5796	758	2756	18	20.3	5804
MM500 Sites	765	2201	18	20.1	23295	706	2276	19	21.1	23023
Adero-Served	1085	2435	19	18.0	362	622	2233	10	15.9	268
Akamai-Served	794	2289	19	21.1	17270	642	2455	20	23.5	7506
Digisle-Served	762	1463	20	13.5	665	847	2368	22	21.9	559
Speedera-Served	815	1859	9	10.6	1297	860	2199	10	11.7	1073

TABLE IV
INSTANTIATION OF CANONICAL PAGE FOR CDNS AND ORIGIN SITES

	Site	Source of Images	Avg. difference in byte size
CDN	Adero	images.mothernature.com	32.8
	Akamai	ivillage.com(a820.g.akamai.net)	37.8
	Clearway	nothingness.org	11.6
	Digisle	fp.cnbc.com	49.8
	Fasttide	www.itat.com	78.0
	Speedera	yack.speedera.net	9.6
US	amazon.com	www.amazon.com	7.6
	bloomberg.com	www.bloomberg.com	29.9
	cnn.com	www.cnn.com	2.9
	espn.com	espn.go.com	1.8
	mtv.com	www.mtv.com	2.1
	nasa.gov	www.hq.nasa.gov	6.6
	playboy.com	www.playboy.com	1.2
	sony.com	www.spe.sony.com	12.0
	yahoo.com	us.yimg.com	10.4
International	UK	www.bbc.co.uk	2.0
	Korea	image.hanmail.net	7.3
	UK	www.msn.co.uk	19.9
	Australia	www.telstra.com.au	11.0
	Brazil	www.uol.com.br	1.2
	Japan	st6.yahoo.co.jp	10.5

1. For CDNs using only DNS redirection, perform a DNS lookup of the server name to obtain an IP address for the server. Record the time taken using *dig*. For CDNs using URL rewriting, we first go to the origin site that is the source of the CDN-served images to determine the current CDN server from which to download the images. If this server is an IP address, then use that address, otherwise do a DNS lookup for the IP address. Note that the DNS lookup time is amortized across *all* images to factor in DNS caching.

2. Retrieve all images from the server at the given IP address. We use *httperf* [11], modified to allow specification of a specific IP address, for all Web object retrievals. This step is not timed, rather, it is intended to ensure that all images have been retrieved and cached by the CDN server if not already present. We want this instance of our study to measure delays for downloading content from the CDN server to the client, and not to unknowingly include delays for a CDN server to retrieve contents from the origin site. By selecting all CDN-served content from the same origin

site, any CDN optimizations for that site should also be available.

3. Retrieve all images from the server at the given IP address using a separate TCP connection for each image with up to four images being retrieved in parallel. Measure the delay to establish the connection, to receive the first byte of the reply for each image request sent, and to retrieve the remaining bytes of each image. In addition, to parallel retrievals using HTTP/1.0 style connections, we also test two retrieval approaches based on HTTP/1.1. In both HTTP/1.1 approaches we use up to two persistent connections to a server. In one test we use serialized requests over these persistent connections and in the other test we used pipelined requests. Not all CDNs supported these options, but results are shown for the CDNs that do support them.

This basic methodology is repeated on a periodic basis over the course of a day so that time–of-day effects will tend to average out. We repeat the test every 30 minutes (with up to 10 minutes of jitter to avoid synchronization effects) for each client to each CDN and each origin site in our test set.

D. Client Sites for Study

The methodology is defined independent of particular clients, CDNs, or origin sites. We exercised the methodology from a collection of worldwide sites that comprise part of the NIMI measurement infrastructure [12]. NIMI consists of a number of widely deployed measurement "platforms" that accept authenticated requests to schedule measurements (in our case, scripts running *httperf*) for some future time, perform the measurements at the indicated time, and send back the results. Approximately two dozen available NIMI platforms were successful in running the measurement for each test set (not all of the 30+ NIMI platforms were available or supported the software that needed to be run). While this is a good number of client sites, they are centered around U.S. university and government laboratory sites—particularly on the U.S. East and West coasts. The narrow domain of our client sites is a limitation of the study.

The methodology of the study is to directly send HTTP requests to CDN and origin servers bypassing any client site proxy servers. All NIMI client sites that successfully ran the scripts were used for testing with no bias towards selecting clients based on their relative proximity to CDN servers. Ideally, a cross-section of the client population would be chosen independent of client proximity to content servers. We do not have data on potential company-installed caches at the client sites which could bias performance measurements.

V. Results

In the study, we took 20,000 measurements from the NIMI clients on three separate weekdays in September, 2000, using only the HTTP/1.0 protocol with parallel retrievals. Four CDNs employing the DNS redirection technique were tested. Later, we took sets of 50,000-90,000 measurements from the NIMI clients on three separate weekdays in January 2001, using the three separate retrievals methods as described in Section IV. In the January tests, we also add two more CDNs to our study—Clearway and Fasttide, which both use the URL rewriting technique. Our initial set of test CDNs only used DNS redirection. All runs started shortly after 4AM EDT and ran every 30 minutes until 1AM or 2AM the following day. The datasets ranged from 24 to 25 NIMI clients returning measurement results. The primary results in this paper are from the last of these three January 2001 datasets, which included results from 19 U.S.-based clients. We focus on U.S. clients because they are more representative of the geographic United States relative to the sparse representation of the international NIMI clients. Comparisons with results from one of the datasets taken from September 2001 are made as appropriate. Mostly, the three experiments from each timeframe show consistent results, but we note differences as appropriate.

In examining the results of our extensive study, we must also note its limitations. As described in Section IV-D, the types of client sites are relatively narrow in scope. We note that the specific results of an empirical study such as this one may change over time—the performance of one CDN company changed dramatically during the few months period of our study. Ideally, the study would be repeated over a periodic time interval to study how specific results change. As a consequence of these limitations, we do not view these results as conclusive in terms of the relative CDN performance. However, one more general conclusion is drawn about CDN use of the DNS mechanism for load balancing.

A. CDN Techniques and Server Use

The specific policies and algorithms used by each CDN are proprietary. However, we can examine the performance results in light of the technique used to direct clients to a server and the number of servers employed by a CDN. Of the six CDN companies we studied, Akamai, Digital Island and Speedera provided partial-site DNS redirection (DR-P), Adero provided full-site DNS redirection (DR-F), Clearway provided URL rewriting (UR) and Fasttide provided a combination of URL rewriting and DNS redirection (URDR).

As part of collecting the performance results, we analyzed the number of distinct IP addresses returned to our clients in response to DNS queries or URL rewriting. This approach allowed us to see the variation in the server selected for individual clients and to determine the total number of servers available for selection. The mean, median, and maximum number of IP addresses used for a CDN on a per-client basis for both the September and January tests is shown in Table V along with the technique. It also shows the total number of distinct IP addresses used by the collective set of clients in each test.

The results show a significant change in the number of servers discovered for Speedera in the two test sets. The size of Akamai's discovered network also grew while the size of the network for Adero and Digital Island was relatively static. The CDN companies using partial-site DNS redirection had the three largest discovered networks in our study. However, there is no clear correlation between the choice of technique used and the number of servers in the CDN network.

B. Response Time Results

The first set of results examine the DNS lookup time to obtain the IP address of a specific CDN server and to download the 18 images from the canonical test page for each of the CDNs and origin sites in our study set. The completion time seen by a client is the sum of these two components. Table VI shows the results of this study for each CDN for a September 2000 and January 2001 test. The results shown are for downloads done with parallel HTTP/1.0 requests. The table also shows combined results for the set of U.S. origin sites and international origin sites.

The mean, median and 90th percentile results show that in September 2000 most CDNs provided better download performance for the U.S. clients than did the U.S. origin sites, and that in January 2001 all CDNs provided substantially better download performance. Download results for Speedera changed dramatically as it performed the worst in the September test, but the best in the January test. This performance improvement corresponded with a large increase in the number of CDN servers used by the NIMI clients in downloading Speedera-served content as shown in Table V. The overall download time results are relatively consistent over the three NIMI tests within each test period.

Clearway incurs no DNS lookup time because load balancing among servers is done by rewriting URLs to directly include server IP addresses. Particularly striking in the DNS results are the large times for Adero with a mean of over five seconds in September and four seconds in January. Further investigation shows that relatively short DNS TTLs not only for the first-level name server of Adero (10 sec.), but also for upper-level name servers for Adero and `images.mothernature.com` (30 min. to 3 hrs.) leads to potentially four non-cached DNS lookups, each of which may introduce a timeout of 5 seconds. Approximately 25% of the DNS lookups took more than five seconds in the January test. The high DNS lookup times for Adero were consistent in all NIMI tests during both the September and January tests.

The three companies using partial-site DNS redirection generally provided the best download results, but these CDNs also had the largest discovered network of servers as shown in Table V. Adero is the only CDN using full-site DNS redirection, but it is not clear whether the DNS timeout problem is inherent in the technique or caused by the DNS TTL settings for the particular origin server being replicated. Clearway, the only CDN using URL rewriting, exhibited a small network of only six servers with most client sites being directed to each of the six servers during the course of the study. However, for its size it exhibited competitive download performance with the other CDNs without incurring a DNS lookup cost.

In the January test we also examined the download times using HTTP/1.1 persistent connections. All CDNs support persistent connections even though Akamai and Digital Island claim to support only HTTP/1.0. The results in Table VII show that all CDNs, except for Speedera with pipelining, successfully supported both serial and pipelined requests. Only 50% of the U.S. and international server sites supported pipelining. The download results in Table VII can be compared with those results in Table VI and show that use of persistent connections yields better results than parallel-1.0 requests for all CDNs. Akamai provides the best overall download times using persistent connections.

We also examined performance for individual clients by using both mean and median download times for each of the CDNs at each of the U.S. clients used during that test. We ignore the DNS lookup delay for this comparison to focus on the quality of the server decision rather than the time to make the server decision. Results in Figure 1 show a cumulative distribution function for the difference between a CDN's performance and the best performing CDN at each client. Results are shown for both test periods. In September, the performance of Digital Island was the best of the CDNs for over 30% of the clients and was within 0.5 seconds of the best for over 80% of the clients. Akamai exhibited the most inconsistency, with over 70% the results either the best or within one second of the best, but also over 10% of its mean and median results more than four seconds slower than the best. The results also show

TABLE V
NUMBER OF DISTINCT IP ADDRESSES RETURNED TO A CLIENT

CDN(Tech.)	Sept. 2000 Mean	Sept. 2000 Med.	Sept. 2000 Max	Sept. 2000 Total	Jan. 2001 Mean	Jan. 2001 Med.	Jan. 2001 Max	Jan. 2001 Total
Adero(DR-F)	4.6	5	9	13	4.8	5	8	11
Akamai(DR-P)	5.8	2	17	65	8.5	8	19	103
Clearway(UR)		–			5.6	6	6	6
Digisle(DR-P)	2.7	3	5	24	3.4	4	6	24
Fasttide(URDR)		–			8.7	9	11	23
Speedera(DR-P)	1.5	1	3	3	10.3	10	26	83

DR-F: full-site DNS redirection, DR-P: partial-site DNS redirection
UR: URL rewriting, URDR: URL rewriting and DNS redirection

TABLE VI
DNS AND DOWNLOAD TIMES (SEC.) FOR IMAGES 1-18 FROM U.S. CLIENTS TO CDNS AND ORIGIN SITES WITH PARALLEL HTTP/1.0 REQUESTS

CDN(Tech.)/ Origin Site	Sept. 2000 DNS Mean	DNS Med.	DNS 90%	Parallel-1.0 Download Mean	Download Med.	Download 90%	Jan. 2001 DNS Mean	DNS Med.	DNS 90%	Parallel-1.0 Download Mean	Download Med.	Download 90%
Adero(DR-F)	5.68	0.14	13.44	1.66	1.27	3.04	4.26	0.15	8.53	1.16	1.02	1.77
Akamai(DR-P)	0.22	0.04	0.20	2.40	0.81	4.79	0.10	0.03	0.17	1.06	0.34	3.01
Clearway(UR)		–			–		0.00	0.00	0.00	1.26	0.85	2.98
Digisle(DR-P)	0.18	0.04	0.14	1.35	0.43	3.08	0.12	0.04	0.15	1.15	0.50	1.80
Fasttide(URDR)		–			–		0.56	0.11	0.51	1.55	0.96	3.37
Speedera(DR-P)	0.14	0.03	0.14	2.70	2.92	4.35	0.15	0.04	0.17	0.57	0.20	1.14
U.S. Origin	0.13	0.00	0.14	2.66	1.25	5.38	0.33	0.03	0.20	3.40	1.06	4.90
Intl Origin	0.51	0.00	0.63	5.67	3.70	11.81	0.46	0.00	0.41	3.62	3.12	5.55

TABLE VII
DOWNLOAD TIMES (SEC.) FOR IMAGES 1-18 FROM U.S. CLIENTS TO CDNS AND ORIGIN SITES WITH SERIAL AND PIPELINED HTTP/1.1 REQUESTS OVER PERSISTENT CONNECTIONS

CDN(Tech.)/ Origin Site	Jan. 2001 Serial-1.1 Download Mean	Serial Med.	Serial 90%	Pipeline-1.1 Download Mean	Pipeline Med.	Pipeline 90%
Adero(DR-F)	0.87	0.81	1.48	0.88	0.75	1.67
Akamai(DR-P)	0.61	0.24	1.36	0.47	0.16	1.07
Clearway(UR)	0.96	0.87	1.61	0.55	0.46	0.81
Digisle(DR-P)	1.13	0.36	1.42	0.51	0.20	0.66
Fasttide(URDR)	1.05	0.85	2.18	0.75	0.55	1.55
Speedera(DR-P)	0.62	0.38	1.28		no support	
U.S. Origin	1.96	1.11	3.84		partial support	
Intl Origin	3.76	3.50	5.72		partial support	

that each CDN provided the best mean or median results for at least one client. Most of the CDNs performed better than the best performing origin server (OS-Best) and the cumulative performance of all origin servers (OS-Cum).

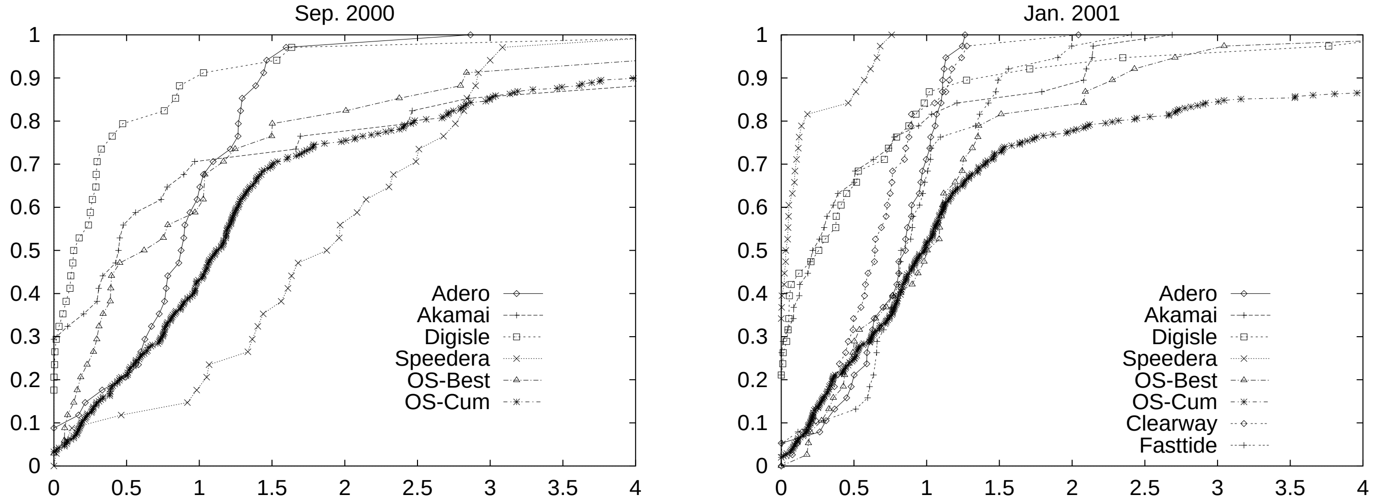

Fig. 1. CDF of Difference Between CDN Mean/Median and Best CDN Mean/Median for Each Client (sec.) with Parallel-1.0 Requests (Sept. 2000 and Jan. 2001)

TABLE VIII

CDN MEAN DOWNLOAD PERFORMANCE RANGE (SEC.) FOR DIFFERENT NUMBERS OF IMAGES AND PROTOCOL OPTIONS (JAN. 2001)

Protocol Option	Number of Downloaded Images			
	6	12	18	54
Parallel-1.0	0.26–0.76	0.40–1.23	0.58–1.53	1.49–3.31
Serial-1.1	0.27–0.53	0.42–0.81	0.61–1.13	1.46–2.52
Pipeline-1.1	0.26–0.50	0.37–0.67	0.47–0.88	1.09–2.04

In the January results of Figure 1, Speedera clearly shows the best results and all CDNs perform relatively better than OS-Cum. Further examination of results for serial-1.1 (not shown) show that Akamai and Speedera both perform well with each having mean and median download times within one second of the best for over 90% of the clients. Akamai was the best CDN performer for the pipeline-1.1 results (also not shown) with all CDNs that support pipeline-1.1 performing better than the cumulative origin server performance. Using pipeline-1.1, Clearway provides consistently good results with all Clearway clients experiencing mean/median downloads within 0.7 seconds of the best performer at each client.

We examined the possible bias in the results due to close proximity of CDN servers to our set of clients. However, as shown in Figure 1, there is not a consistent best performer among the CDNs. All CDNs exhibited best performance for at least one client in the September 2000 results with Akamai, Digital Island and Speedera each giving the best performance for about 30% of the clients in the January 2001 results. Thus the results do not show a systematic bias in favor of a particular CDN.

We also examined variation in results due to the number of images to be downloaded. A summary of these results (from January 2001 test) are given in Table VIII, which shows the range of mean download times for CDNs for a given number of images to download and each protocol option. The results show close to linear correspondence between the number of images and the download performance for each of the protocol options. Reducing the number of images and using the HTTP/1.1 protocol options both reduce the range variation among the CDNs. In these cases the DNS lookup performance becomes a bigger contributor to the overall response time. Download times from origin servers are not shown in Table VIII, but are higher than the CDN range. For example, for serial-1.1, the mean download times from U.S. origin servers are 1.06, 1.46, 1.96 and 4.87 seconds for 6, 12, 18 and 54 images. These results indicate CDNs offer better overall performance than this set of origin servers for this set of clients. The results indicate that caching of these images, which reduces the number needing to be retrieved, reduces, but does not eliminate, the performance difference.

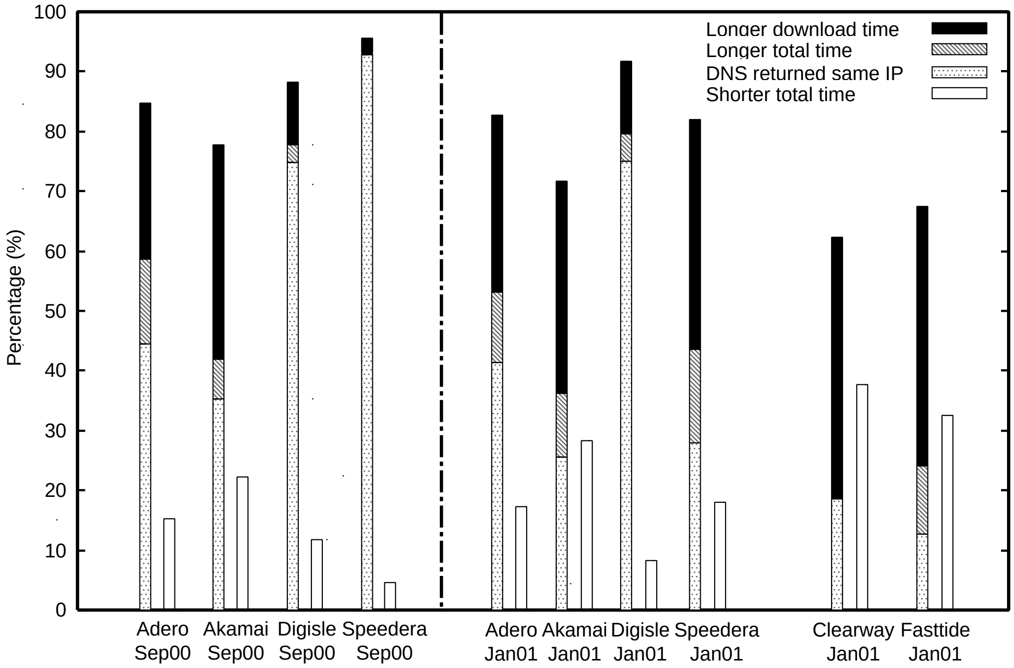

Fig. 2. Performance Comparison of New vs. Fixed IP Address

TABLE IX
PARALLEL-1.0 PERFORMANCE (SEC.) FOR SERVER AT NEW AND FIXED IP ADDRESSES (JAN. 2001)

CDN (DNS TTL in sec.)	New Download Time			New Completion Time			Fixed IP Download Time		
	Mean	Med.	90%	Mean	Med.	90%	Mean	Med.	90%
Adero (10)	1.15	1.02	1.73	5.40	1.39	9.60	1.09	0.51	1.60
Akamai (20)	1.06	0.34	3.01	1.15	0.39	3.05	1.00	0.41	3.00
Clearway (N/A)	1.19	0.84	2.94	1.19	0.84	2.94	1.16	0.76	3.07
Digisle (20)	1.19	0.47	1.83	1.31	0.52	2.30	1.21	0.43	1.70
Fasttide (230)	1.58	0.96	3.37	2.10	1.19	4.72	1.46	0.91	3.25
Speedera (120)	0.57	0.20	1.18	0.72	0.26	1.53	0.53	0.18	1.01

C. DNS Load Balancing

A basic question to ask regarding CDNs that use DNS redirection is: What are the benefits versus the costs of using DNS load balancing? CDNs assign small DNS TTLs for the IP addresses they return so that clients are obligated to do frequent DNS lookups. This approach gives CDNs more control over which of their servers clients can use. We observed authoritative DNS TTLs of 10 seconds for Adero, 20 seconds for Akamai, 20 seconds for Digital Island, 120 seconds for Speedera and 230 seconds for Fasttide. Comparing these authoritative DNS TTLs to those used by a selected set of popular origin sites we find the origin site DNS TTLs ranged from 15 minutes for `cnn.com` to six hours for `espn.com`, except for a one minute DNS TTL for `bloomberg.com`.

To examine the effectiveness of DNS load balancing in yielding better download and completion times for clients we modified our basic testing structure so that each client does a DNS lookup and stores a "fixed" IP address for each CDN server. This fixed address was actually looked up every eight hours, asynchronously to our other testing. We compared how often the fixed IP address was the same as the new one obtained from the DNS for the test each half-hour (the lookup was unnecessary overhead). If the IP addresses were different, we did a separate preload (step 2 in Section IV-C) and download (step 3) from the fixed IP address and compared the download results obtained from the two separate servers, in order to assess just what the redirect gained. We show a summary of these results for parallel-1.0 requests in Figure 2 for both the September

2000 and January 2001 tests. Results for the HTTP/1.1 protocol options in January are similar in nature.

The results for each CDN are broken into four categories. The first three are plotted together, and represent cases in which the extra DNS lookup had no positive benefit; the last represents the case where the redirection was clearly beneficial and is plotted in a separate column. In the Sep. 2000 test the first category (fixed and new IP address are the same) accounts for 30–40% of the cases for Akamai to over 90% of the cases for Speedera. This category accounts for 15% (Fasttide) to 70% (Digital Island) in the January test. In these cases, the download times would be identical for the fixed and new IP address, but DNS lookup costs are incurred for the new IP address, increasing the overall completion time. The second category represents cases where the combined DNS and download time for the new IP address are larger than the download time for the fixed IP address, but the download time by itself is not. Thus, we lost performance in this case, but only when we factor in the DNS overhead. This category represents up to 10% of the cases for both tests. The third category of comparison occurs when the download time (irrespective of DNS costs) for the new IP address is larger than for the fixed IP address. This category represents a clear loss of performance, even if we do not consider the cost of the DNS lookup. Akamai has the most cases in this category in both tests (30-40%).

The last category, plotted separately, shows the percentage of cases where the overall completion time is better for the new IP address over the download time of the fixed IP address. These cases show where the time to do a DNS lookup is warranted in terms of better overall response time for the client. In September, Speedera had 5% of its cases in this category with about 20% for Akamai. In January, Akamai, Clearway and Fasttide were in the 30-40% range.

Table IX shows the mean, median and 90th percentile values for the new download, new completion (including DNS lookup), and fixed download times for the parallel-1.0 requests in January 2001. The results are mixed as to whether the average download times for the new IP address are better than for the fixed IP address. However, if we compare the completion time for the new IP address with the download time for the fixed IP address, which incurs no DNS cost, then we see that for all CDNs, except median Akamai times in January 2001, the fixed IP address performs better. Furthermore, the 90% results indicate that the DNS lookup is not improving the worst case download times. Lowering the bound on worst case results is another argument for using small DNS TTLs, but only in the case of Clearway are the 90% download results better for the new versus the fixed IP address. Inclusion of the DNS lookup times only increases the difference in results. In results not shown for the two HTTP/1.1 protocol options the only case where the completion time for the new IP address is better than the fixed download time is for median serial-1.1 results of Speedera.

We performed a similar study and analysis using the previous IP address returned in our study rather than a fixed IP address. In this study, the previous server was obtained in a DNS lookup that occurred 30 minutes ago rather than up to eight hours ago. In the September test, for all of the CDNs, except Digital Island, the download time using the new IP address was actually worse than the download time using the previous IP address and for all of the CDNs, the completion time with the new IP address was worse than the download time with the previous IP address. In the January test, the completion time using the new IP address was always worse than the download time using the previous IP address except for the Clearway median results under serial-1.1. Results from other test sets were similar.

These results indicate that use of a small DNS TTL by the CDNs, which forces a DNS lookup in the critical path of resource retrieval, does not generally result in better server choices being made relative to client response time in either average or worst case situations. In addition, the download time from a previously selected server is often better than from the download time from the newly selected server. These results indicate that the CDN servers are generally not loaded so frequent DNS lookup costs to select from the set of servers does not result in a performance improvement. Rather, it makes sense for CDNs to increase the DNS TTL given to a client unless the servers are known to be loaded.

VI. Ongoing and Future Work

We are extending our study to include delivery of streaming media content. Recent work [13] shows that although streaming media objects are a small fraction of the number of resources, they contribute a significant fraction of the bytes. We used two proxy log sets discussed in Section III-B to look for streaming media objects served by CDNs. Our approach differs from [13] by looking only for CDN-served objects versus such objects served by all servers. However our results are similar to [13] in that we found less than one percent of CDN-served objects were for streaming media, but these objects accounted for 14-20% of the bytes served by CDNs over HTTP.

Given the presence of streaming content, an important direction of ongoing and future work is to understand how CDNs are being used and how they perform for serving this content. One piece of ongoing work is use the advance search feature of a popular search engine to extract Web pages that *refer* to URLs containing suffixes associated with streaming content. While not all of these URLs are still available, we can use appropriate methods to gather meta-information about these objects.

We are also working to extend our testing methodology to streaming media content. For example, many of the Microsoft Media Services (MMS) files are accessible via HTTP and attributes such as the length of the session can be controlled via HTTP headers (e.g. `Pragma: max-duration`). We can then employ the testing methodology to download streaming content of a specific duration and compare the actual time of the download to the expected.

While this simple extension to our testing methodology can be used for measuring download performance for some streaming content, we need to consider other factors as well. These factors include jitter, loss rate, and achieved frame rate. We also need to consider other protocols (e.g., RTSP) and encoding formats.

VII. Summary

This paper provides a timely discussion and analysis of CDNs. We have used multiple data streams: active measurements obtained via repeated crawls over a period of time and passive measurements representing large number of users from different organizations. We have also analyzed content types commensurate with traffic patterns on the Web. The primary performance study has been repeated more than once. Using the results of our work we reexamine the research questions posed at the beginning of this paper.

Looking at the extent to which CDNs are being used to serve popular content we found that in December 2000 31% of one list of 127 popular Web sites and 17% of a larger list of 1030 popular sites use a CDN to serve content. Compared to 1-2% for 670 popular Web sites in November 1999, we see a clear increase in the number and percentage of popular origin sites using CDNs.

A periodic crawl of home pages of popular Web sites found that 86-94% of CDN-served objects were seen in a previously retrieved version of a home page indicating that the set of CDN-served URLs changes little. Less than one percent of the content of these URLs was found to change. Using Web proxy logs we found that requested objects served by CDNs are largely images with a 20-30% higher cache-hit rate than for non-CDN-served images.

A significant contribution of this work is a methodology that can be applied without bias for measuring the client-perceived performance of retrieving content from CDNs or origin servers. The definition of a canonical page consisting of a fixed number of objects of a defined sizes can be instantiated based upon the set of objects available from a CDN or origin server.

We applied the testing methodology from a set of two dozen client sites over a period of many months. In the results each CDN yielded the best performance for at least one client when considering mean and median download time as measures of comparison. Some CDNs generally provide better results than others when we examine results from the entire set of clients. The performance of one CDN company clearly improved between the two testing periods in our study due to a dramatic increase in the number of distinct servers employed in its network.

In looking at relative performance with a selected set of popular U.S. origin servers, we found that in our initial set of tests, the CDNs performed better, but not significantly better than the origin servers. In the latter set of tests the CDNs performed significantly better for downloading small to large numbers of images. We stress that our client sites are centered around two dozen academic and laboratory sites, and constitute a relatively narrow client mix.

Despite these difficulties on making broader conclusions, our methodology should be used by content providers seeking to evaluate the potential of CDN performance by carrying out a similar study from a cross-section of their own customer sites to better understand which CDNs will provide better response times relative to servers at the content provider site.

The results of the study allows us to draw conclusions about the use of small DNS TTL values by CDNs to load balance amongst a set of servers. We compared the download time for the set of images from the server returned for a DNS lookup with the download time for a fixed server of the CDN. We also looked at results using the previous server returned by a DNS lookup. In most cases we found that the download time for the newly obtained server was not better than for the fixed or previous server. In almost all cases, for all CDNs and all HTTP protocol retrieval options, we found that when factoring in the time for the DNS lookup, response time was actually better using the previous or fixed server. The results indicate that even worst-case client response time is generally not improved with a DNS lookup to find a new server.

VIII. Acknowledgements

We would like to thank Vern Paxson for his earlier role in this project and for his help in obtaining data via the NIMI infrastructure. We also thank the reviewers for their suggestions.

REFERENCES

[1] B. Krishnamurthy, J. C. Mogul, and D. M. Kristol, "Key Differences between HTTP/1.0 and HTTP/1.1," in *Proc. Eighth International World Wide Web Conference*, (Toronto), May 1999. `http://www.research.att.com/~bala/papers/h0vh1.html`.

[2] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," in *Proceedings of the IEEE Infocom 2001 Conference*, (Anchorage, Alaska USA), Apr. 2001.

[3] B. Krishnamurthy and C. E. Wills, "Analyzing factors that influence end-to-end web performance," in *Proceedings of the Ninth International World Wide Web Conference*, (Amsterdam, Netherlands), April 2000.

[4] "Media metrix." `http://www.mediametrix.com`.

[5] "100hot.com." `http://100hot.com/`.

[6] B. Krishnamurthy and M. Arlitt, "PRO-COW: Protocol Compliance on the Web—A Longitudinal Study," in *Proceedings of Usenix Symposium on Internet Technologies and Systems, USITS 2001*, Mar. 2001. `http://www.research.att.com/~bala/papers/usits01.ps.gz`.

[7] NLANR, "Proxy cache log traces," 2001. `ftp://ircache.nlanr.net/Traces/`.

[8] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek, "The measured performance of content distribution networks," in *Proceedings of the 5th International Web Caching Workshop and Content Delivery Workshop*, (Lisbon, Portugal), May 2000. `www.terena.nl/conf/wcw/Proceedings/S4/S4-1.pdf`.

[9] M. Koletsou and G. M. Voelker, "The medusa proxy: A tool for exploring user-perceived web performance," in *Proceedings of the 6th International Web Caching Workshop and Content Delivery Workshop*, (Boston, MA), June 2001. `http://www-cse.ucsd.edu/~voelker/pubs/medusa-wcw01.pdf`.

[10] B. Krishnamurthy and J. Rexford, *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*. Addison-Wesley, May 2001. ISBN 0-201-710889-0.

[11] D. Mosberger and T. Jin, "httperf—a tool for measuring Web server performance," in *Proceedings of the First Workshop on Internet Server Performance (WISP '98), Madison, WI*, pp. 59–67, June 1998. `http://www.hpl.hp.com/personal/David_Mosberger/httperf`.

[12] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An architecture for large-scale Internet measurement," in *IEEE Communications*, August 1998. `ftp://ftp.ee.lbl.gov/papers/nimi-ieee-comm98.ps.gz`.

[13] M. Chesire, A. Wolxman, G. M. Voelker, and H. M. Levy, "Measurement and analysis of a streaming-media workload," in *Proceedings of Usenix Symposium on Internet Technologies and Systems (USITS 2001), San Francisco, CA*, Mar 2001.