# A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers

Zhuoqing Morley Mao*, Charles D. Cranor, Fred Douglis†, Michael Rabinovich,
Oliver Spatscheck, and Jia Wang
*AT&T Labs–Research*

## Abstract

Content Distribution Networks (CDNs) attempt to improve Web performance by delivering Web content to end-users from servers located at the edge of the network. An important factor contributing to the performance improvement is the ability of a CDN to select servers in the proximity of the requesting clients. Most CDNs today use the Domain Name System (DNS) to make such server selection decisions. However, DNS provides only the IP address of the client's local DNS server to the CDN, rather than the client's IP address. Therefore, CDNs using DNS-based server selection assume that clients are "close" to their local DNS servers.

To quantify the proximity between clients and their local DNS servers, we propose a novel, precise, and efficient technique for finding the associations of client to local DNS servers. We collected more than 4.2 million such unique associations in three months. From this data, we study the impact of proximity on DNS-based server selection using four different proximity metrics. We conclude that DNS is good for very coarse-grained server selection, since 64% of the associations belong to the same Autonomous System. DNS is less useful for finer-grained server selection, since only 16% of the client and local DNS associations are in the same *network-aware cluster* [13] (based on BGP routing information from a wide set of routers). As an application of this methodology, we evaluate DNS-based server selection in three of the largest commercially deployed CDNs to study its accuracy.

## 1 Introduction

Creating and managing a high-performance, Internet-scale Web service is a formidable challenge involving deployment of multiple Web servers in strategic locations throughout the network. The introduction of Content Distribution Networks (CDNs) has allowed organizations to overcome this challenge by outsourcing the distribution of their Web content. With CDNs, content providers need only to supply an origin Web server — the CDN distributes the content to end users through a set of CDN servers it has deployed in the network. Ideally, this reduces Web response time and download latencies in addition to providing overload protection and bandwidth savings.

In a well-designed CDN, servers are placed to avoid congested links and slow network paths. When a Web client requests content, the CDN dynamically chooses a server to route the request to, usually one that is appropriately close to the client. Note that this dynamic CDN request routing is an extra step that is not necessary for stand-alone Web servers. Efficient CDN server selection allows CDNs to overcome the extra overhead of the dynamic routing step by taking advantage of improved connectivity to the end user. CDN server selection applies for both static and dynamic content. In the latter case, content can be dynamically assembled at the edge servers [1].

CDNs typically perform dynamic request routing using the Internet's Domain Name System (DNS) [11]. The DNS is a distributed directory whose primary role is to map fully qualified domain names (FQDNs) to IP addresses. To determine an FQDN's address, a DNS client sends a request to its local DNS server. The local DNS server resolves the request on behalf of the client by querying a set of authoritative DNS servers. When the local DNS server receives an answer to its request, it sends the result to the DNS client and caches it for future queries. Each DNS record has a time-to-live (TTL) field that tells the local DNS server how long it may cache the result.

Normally, an authoritative DNS server's association from FQDNs to IP addresses is static. However, CDNs

*Zhuoqing Morley Mao (email: zmao@cs.berkeley.edu) is a Computer Science graduate student at University of California, Berkeley. This work was done during her internship at AT&T Research Labs.

†Current affiliation: IBM Research

use modified authoritative DNS servers for CDN server selection. The results of a DNS query to one of these DNS servers may vary dynamically depending on factors such as the source of the request and the condition of the network. Typically, the CDN's authoritative DNS server maps the client's local DNS server address to a geographic region within a particular network and combines that with network and server load information to perform CDN server selection. To enable fast reaction to dynamic resource changes, the answer returned by the CDN's DNS server has a small TTL. This approach is largely transparent to the client, and works for any Web content (including both HTML and streaming media).

Although DNS-based server selection is transparent and general, it has two inherent limitations [15, 4]. First, it is based on the implicit assumption that clients are close to their local DNS servers. The CDN DNS server performing dynamic request routing only has access to the client's local DNS server's IP address—it does not know the client's own IP address. However, the assumption that clients are close to their local DNS server may not be valid. For example, the client might be using a local DNS server hierarchy in which the outermost local DNS server that communicates with authoritative DNS servers may be far removed from clients; the client may have been configured with a local DNS server which is far away; or the client may be using a secondary local DNS server that is more distant from it than its primary local DNS server. Therefore, using only the local DNS server information to select CDN servers has the inherent risk of selecting a server farther away from the client than other available CDN servers.

The second inherent limitation of DNS-based server selection is that a single request from a local DNS server can represent differing numbers of Web clients — this is called the *hidden load factor* [8]. The hidden load has implications on a CDN's load balancing algorithm. For example, a DNS request from a local DNS server of a large ISP may result in many more Web requests than a DNS request from a local DNS server of a small site. CDNs need to be able to properly weigh individual DNS requests to distribute Web requests among its CDN servers. If the hidden load factors are known, load balancing algorithms described by Colajanni, et al. [7, 8] can be easily deployed to achieve better load distribution. On the other hand, if the hidden load factors are not known, fine-grained request distribution may be difficult.

We study the extent of the first limitation and its impact on CDN server selection. To this end, we developed a simple, non-intrusive, and efficient mapping technique to determine the associations between clients and local DNS servers. We deployed this technique on several sites to collect an extensive data set which we use to study the impact of proximity on DNS-based server selection using four different proximity metrics. We conclude that DNS is good for very coarse-grained server selection, since 64% of the associations belong to the same Autonomous System (AS). DNS is less useful for finer-grained server selection, since only 16% of clients use DNS servers in the same *network-aware cluster* [13] (based on BGP routing information). We also measure the CDN server distribution of several real-world CDNs to evaluate whether the proximity of a client to its local DNS server leads to potentially suboptimal CDN server selection decisions in practice. Our technique could also be used to determine hidden load factors by associating the HTTP request pattern in the Web server logs with the DNS request information.

Our work makes the following contributions. We developed a novel measurement methodology and architecture for accurately collecting local DNS server IP addresses of Web clients. We demonstrated its successful deployment on several sites including a large commercial site and through the collection of a huge database of associations. Based on this data, we did an extensive analysis of the proximity between clients and their local DNS servers and discovered that significant improvement in proximity is possible by configuring clients to use a closer local DNS server. Finally, we evaluated the impact of the proximity between clients and their local DNS servers on server selection in three of the largest commercially deployed CDNs. We conclude that DNS is good for very coarse-grained server selection, but less suitable for fine-grained request distribution.

The rest of the paper is organized as follows. Section 2 describes our methodology and measurement setup for gathering DNS client associations. In Section 3, the association results are analyzed in detail to evaluate the proximity between the client and its local DNS server. Then, in Section 4 we study the impact of proximity evaluation on DNS-based server selection in three of the largest commercially deployed CDNs. Related work is covered in Section 5. In section 6, we discuss future work. Section 7 concludes.

## 2 Experimental methodology

In this section we describe our novel technique for determining a Web client's local DNS server. This is a necessary first step in measuring the closeness of clients to their local DNS servers. We also evaluate the impact of our technique on end user performance. Later, in Section 5, we will explain how our technique is a significant improvement over related previous work in terms of efficiency, nonintrusiveness, and accuracy.

## 2.1 Measurement setup

There are three main components necessary to use our technique: a specialized authoritative DNS server, an HTTP redirector, and a one-pixel embedded transparent GIF image. To obtain a client population we solicited volunteer Web sites. All the volunteers had to do to participate in our study was to add a link to our one-pixel transparent GIF to the end of one or more of their commonly accessed Web pages. Assuming the experiment is hosted by us at `example.com`, this involves adding the following HTML code towards the end of a web page:

```
<img src="http://xxx.rd.example.com/tr.gif"
   height=1 width=1>
```

To allow us to easily account for hits from different sites, each participant replaces `xxx` in the URL with a site identifier[1]. This allows us to easily add additional volunteer sites without having to make any changes to our Web or DNS server configuration.

When a Web client loads the one-pixel embedded image, our technique allows us to match the address of the local DNS server resolving host names on behalf of the client with the address of the client itself. This process is shown in Figure 1. First, the client attempts to get the image from `xxx.rd.example.com` — our HTTP redirector. Rather than serving the image, the redirector determines the client's IP address and issues an HTTP redirect to `ipCLI.cs.example.com`, where `CLI` is replaced with a string encoding the IP address of the client (step 2). Next, the client contacts its local DNS server to resolve this domain name (step 3). The client's local DNS server attempts to resolves `ipCLI.cs.example.com` by sending a DNS request to our authoritative DNS server (step 4). At this point our authoritative DNS server logs the IP address of the local DNS server and the client IP address embedded within the query. It then sends the address of the content server hosting the image back to the client's local DNS server (step 5). This resolution is passed on to the client (step 6), which retrieves the image from the content server (steps 7 and 8).

[1] Our authoritative DNS server [6] allows host names to be wildcarded, so we can set an address for `*.rd.example.com`.

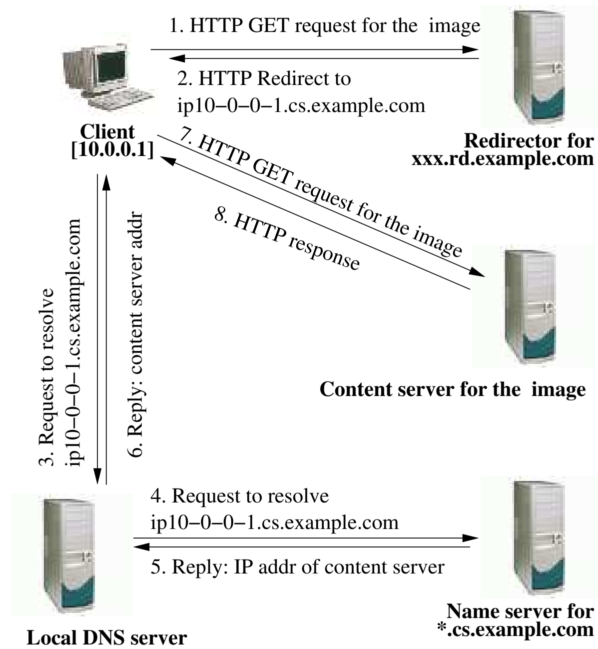


Figure 1: Embedded image request sequence

This measurement methodology has a limitation for clients that do not fetch inlined images and those that abort the page download process before the DNS resolution is made for the embedded image. In these cases, we are unable to collect their local DNS server information.

Note that in some cases, a local DNS server *hierarchy* may exist. The local DNS server recorded in our measurement is the outermost local DNS server which directly contacts the authoritative DNS server for the `example.com` domain. In DNS-based server selection, the CDN's DNS server only sees the outermost local DNS server. In this study, this outermost DNS server is what we refer to as the "local DNS server."

This measurement approach is fully deterministic. It collects one association each time a new client visits a site with the embedded image. Multiple pages on the same site, or subsequent visits to the same page, may result in repeated retrievals of the calibrating image depending on the client's caching policy.

Note that the redirector also logs client requests — this information can be correlated with the DNS and web server logs to obtain the hidden load factors. Statistics on client browsing characteristics can also be gathered from the HTTP headers in the redirector log.

Table 1: Keynote image overhead measurements

| Location | Avg download latency (sec) | | Increased |
|---|---|---|---|
| | without image | with image | overhead |
| World wide | 1.17 | 1.31 | 12% |
| US | 1.04 | 1.14 | 10% |

## 2.2 Measurement impact

Because we propose to use our measurement infrastructure on a production Web site, it is important to evaluate its impact on the server performance and other aspects of its operation. The additional overhead our measurement technique imposes on Web client performance is the retrieval of the transparent image, including the HTTP redirect and extra DNS requests. Because the image is transparent, it does not visually affect the page. Furthermore, the image is small in size—43 bytes—which keeps the added delay to a minimum. We also encourage participants to include the image at the end of the HTML page containing it; therefore, browsers will normally request it last. Thus, the extra latency associated with the image is usually hidden from the user's Web browsing experience. Another advantage of the small size of the image is that when the image is not available for download, it does not affect the visual appearance of the Web page at all.

Our custom HTTP redirector is a single-threaded, non-blocking, 300-line C program. The redirector responds to all Web requests with a "302 Moved Temporarily" HTTP redirect to a URL with the client's IP address embedded in it. Due to the small size and overhead of the redirector, we found it to be highly reliable and more responsive than a standard Web server.

To validate the claim of a small increase in latency, we measured a simple Web page with Keynote [2] to compare the download time with and without the embedded calibrating image. Keynote probes are located in 25 cities within the US and 10 cities outside the US. The Web page we measured had a total size of 39 Kbytes including 13 images and was accelerated by a CDN. The increased overhead percentage is therefore higher than we would expect for a regular unaccelerated Web page with more embedded images. Table 1 shows that the increased overhead averages less than 140 ms, which is 10–12% of the total download time.

We also tested our system to see what would happen in the event of a failure of the redirector, image content server, or DNS server. We found that the impact of failure on the user is minimal. We tested the failure of these three components using Microsoft Internet Explorer (MSIE) 6 and Netscape Navigator 6 and found that those browsers will first load the rest of the Web page and then time out while trying to fetch the image.[2] There is no visible change to the Web page or any pop-up error message; however, the Netscape logo or MSIE browser logo will provide visual feedback until the browser times out.

Table 2: Participating sites in the study

| Site | Type | # of 1-pixel image hits | Duration |
|---|---|---|---|
| 1 | att.com | 20,816,927 | 2 months |
| 2,3 | Personal pages (commercial domain) | 1,743 | 3 months |
| 4 | Research lab | 212,814 | 3 months |
| 5-7 | University sites | 4,367,076 | 3 months |
| 8-19 | Personal pages (university domain) | 26,563 | 3 months |

Table 3: DNS and HTTP log statistics for all sites

| Type | Count |
|---|---|
| Client-LDNS associations | 4,253,157 |
| HTTP requests | 25,425,123 |
| Unique client IPs | 3,234,449 |
| Unique LDNS IPs | 157,633 |
| Client-LDNS associations where client and LDNS have the same IP address | 56,086 |

# 3 Analysis results

We conducted our measurement study for about three months, and nineteen Web sites participated, as described in Table 2. We classify these sites into two categories: *commercial* (sites 1-3) and *educational* (sites 4-19). As we show in Section 3.1, the client and local DNS associations visiting these two sites have very different characteristics. For ease of discussion, we use *LDNS* to represent a local DNS server. A total of 4,253,157 unique client and LDNS associations were collected. Table 3 presents the statistics of the DNS server and the redirector log for all sites.

To study the proximity between the client and its local

[2] We tested with the default setting without any special options. Some older versions of both browsers were also tested giving the same behavior.

DNS server, we use the following four metrics.

- **AS clustering.** Autonomous System (AS) clustering refers to observing whether a client is in the same AS as its local DNS server. An AS is a region under a single administrative control. A single AS might contain an entire backbone or a large corporation which might span multiple continents. Therefore, AS-based clustering is the most coarse-grained metric we use.

- **Network clustering.** This metric observes whether a client is in the same *network-aware cluster* (NAC) as its local DNS server, where network clusters are identified by the *network-aware clustering* technique [13] using prefix entries from BGP routing table snapshots from a wide set of routing tables. *Longest prefix matching* is used to map clients to network clusters identified by a network prefix. All the clients within a network cluster are topologically close together and with a high probability belong to the same administrative domain. Validation tests (in [13]) using *nslookup* and *traceroute* show that the accuracy of network clustering is above 90% across all the Web logs from the study by Krishnamurthy and Wang. Network clustering is much more fine-grained than AS clustering [12].

  For both AS and network clustering, BGP prefixes and the association of IP CIDR blocks to ASes were extracted from an extensive set of BGP tables collected on May 27, 2001 from the sources listed by Krishnamurthy and Wang [13] and Telstra Internet [5]. There are a total of more than 440,000 unique routing entries.

- **Traceroute divergence.** This metric, used previously in [15], is based on the length of divergent paths to the client and its local DNS server from a probe point using *traceroute*. It is defined to be the maximum number of disjoint network hops from a probe location to the client and its LDNS.

- **Round-trip time correlation.** This metric, used previously in both [15] and [4], refers to examining the correlation between the message round-trip times from a probe point to the client and its local DNS server.

AS clustering, network clustering, and traceroute divergence are topology-oriented metrics, while round-trip time correlation is a performance-oriented metric. AS and network clustering are passive, requiring no active probing. The other metrics are highly dependent on the probe locations. To obtain an exhaustive evaluation of proximity, we include all four metrics in our study.

Table 4: Aggregate statistics of AS/network clustering

| Metrics | # of client clusters | # of LDNS clusters | total # of clusters |
|---|---|---|---|
| AS clustering | 9,215 | 8,590 | 9,570 |
| Network clustering | 98,001 | 53,321 | 104,950 |

### 3.1 AS and network clustering

Table 4 shows the aggregate statistics from the data we collected—the number of clusters containing clients, the number of clusters containing local DNS servers, and the total number of clusters. We note that from daily routing table analysis from several major ISPs [9], up to 12,000 unique ASes were identified as being in use on November 12, 2001. The theoretical limit on the possible number of ASes is determined by the 16-bit AS identifier, resulting in a total of 64K ASes. Thus, we observed close to 80% of ASes that were identified on November 12, 2001 and close to 15% of the total possible ASes. With regard to network clusters, the maximum number of network clusters is 440K, since we used 440K unique prefixes. A one day extract from the 1998 Winter Olympic Games server log has 9,853 client clusters [13]. Thus, our measurement data contains close to ten times as many client clusters from one day of a popular Web server log and close to 25% of all possible network clusters. We conclude that the data we collected is extensive and covers a significant number of ASes and network clusters.

Table 5 shows the percentage of client-LDNS associations sharing the same cluster for clients visiting educational sites, commercial sites, and all sites in our measurement study. We observe that clients visiting educational sites have better proximity to their local DNS servers using the network- and AS- clustering metrics. This is expected since most of these clients also come from universities, which generally have a denser distribution of local DNS servers and better local DNS configurations than commercial ISPs. Because the majority of our log results from hits to the commercial sites, the proximity values for clients visiting all participating sites are very close to those visiting commercial sites alone. Because CDNs are most likely to accelerate commercial sites, we believe our client mix is representative

Table 5: Percentage of client-LDNS associations sharing the same cluster classified according to the types of domains visited by the clients

| Metrics | Client IPs | | | HTTP requests | | |
|---|---|---|---|---|---|---|
| | educational | commercial | combined | educational | commercial | combined |
| AS cluster | 70% | 63% | 64% | 83% | 68% | 69% |
| Network cluster | 28% | 16% | 16% | 44% | 23% | 24% |

of clients visiting a CDN-accelerated site. In the following discussion, we consider clients visiting all participating sites.

Using AS clustering, 64% of distinct client-LDNS associations share the same AS. Thus, more than half of the clients use a local DNS server in the same AS. This is expected, since it is common for an administrative domain to run its own DNS server. If users configure their DNS settings correctly, they typically use the LDNS in their administrative domain by default. About 69% of the HTTP requests come from clients using an LDNS server in the same AS cluster. This means clients with LDNS in the same AS are slightly more active than those that use an LDNS in another AS.

The above results indicate that in about 64% of the cases, CDNs could select appropriate servers using DNS redirection with the granularity of ASes. Thus, even if a CDN deployed a cache in every AS in the world, it could select the closest cache according to the AS metric only in 64% of the cases. However, AS clustering does not reveal how well redirection works for finer-grained load-balancing. An AS can span large geographical regions, causing network delays between two hosts within the same AS to be relatively high. For finer-grained load-balancing it is therefore important to consider network clustering, which groups together IP addresses that are close together topologically and likely to be under the same administrative domain.

The observations using network clustering are significantly different from the AS clustering results. Only 16% of the client-LDNS associations are in the same network cluster. This shows that most clients are *not* in the same routing entity as their local DNS servers. If the HTTP request count is taken into account, about 24% of the HTTP requests in our logs originated from clients that used an LDNS in the same network cluster. Again, the difference between these two numbers demonstrate that clients with LDNS in the same network clusters are more active than those with LDNS in a different network cluster.

Overall, these results indicate that DNS-based redirection can confidently select appropriate CDN servers with the granularity of an AS. However, for CDNs with multiple servers in the same AS, the selection may not be as accurate. If there is a CDN server in each network cluster, then DNS-based redirection will only select the CDN server in the same network cluster as the client about 24% of the time.

### 3.2 Traceroute divergence

Another metric to evaluate the proximity between the client and its local DNS server is the maximum number of disjoint network hops from a probe location to the client and its local DNS server. In [15], this metric is referred to as the *traceroute cluster size*. The smaller the cluster size or *traceroute divergence*, the closer the client is to the local DNS server. In many of our traceroute results, we found that the network routes from the probe site to the client and its LDNS diverge and converge multiple times due to router load balancing. We use the last point of divergence as the reference for calculating disjoint network hops. For example, Table 6 shows the network routes obtained by performing traceroute to the client 112.74.197.163[3] and its LDNS 112.25.195.1. We use hop 11 instead of 2 as the point of divergence. Thus, the traceroute divergence in this example is $max(14 - 11, 13 - 11) = 3$.

We selected four probe sites representing candidate CDN servers and performed traceroute to a sample of clients and local DNS servers from the log. The sample consists of 48,908 client-LDNS pairs or 66,975 IP addresses. It is obtained by randomly selecting one client-LDNS pair from the top half of the client network clusters generating the most HTTP requests. The number of client-LDNS pairs reached by an individual probe site ranges from 9,878 to 11,935. In about 20% of these, both the client and the LDNS belong to the same network cluster. And in about 75% of these, both the client and the LDNS belong to the same AS cluster.

[3]For privacy concerns, the IP addresses have been anonymized.

Table 6: Traceroute divergence

| | |
|---|---|
| 1 112.0.1.1 6 ms | 1 112.0.1.1 5 ms |
| 2 112.124.182.17 6 ms | 2 112.124.182.17 15 ms |
| 3 112.123.1.10 7 ms | 3 112.123.1.22 14 ms |
| 4 112.122.1.149 8 ms | 4 112.122.5.246 7 ms |
| 5 112.122.2.173 25 ms | 5 112.122.2.2 24 ms |
| 6 112.122.2.206 32 ms | 6 112.122.2.206 31 ms |
| 7 112.122.2.41 34 ms | 7 112.122.2.41 35 ms |
| 8 112.122.2.26 71 ms | 8 112.122.2.26 68 ms |
| 9 112.122.2.121 75 ms | 9 112.122.2.121 77 ms |
| 10 112.123.145.25 73 ms | 10 112.123.145.25 72 ms |
| **11 112.124.23.6 72 ms** | **11 112.124.23.6 73 ms** |
| 12 112.25.192.2 72 ms | 12 * * * |
| 13 112.25.192.181 73 ms | 13 * 112.25.195.1 71 ms |
| 14 112.74.197.163 92 ms | |

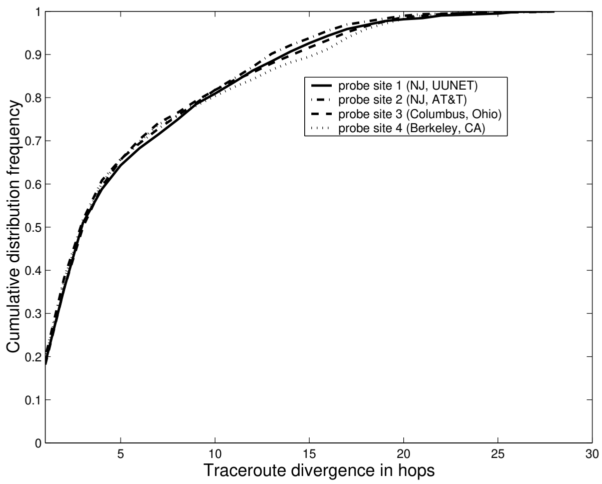


Figure 2: Proximity evaluation using traceroute divergence

Figure 2 shows the cumulative distribution of traceroute divergence for the sampled client-LDNS pairs. About 14% of them have traceroute divergence of 1. The mean divergence varies from 5.8 to 6.2 depending on the probe site, and the median traceroute divergence is 4 from all four probe sites. This means that a large fraction of clients are topologically quite *close* to their local DNS servers using the hop count metric. At most 30% of the client-LDNS pairs have traceroute divergence of size 8. This result is slightly inconsistent with the results described by Shaikh, et al. [15] considering 1,090 client-LDNS pairs of dial-up ISPs. We believe that the difference can be explained by the fact that our results are based on the analysis of a much larger set of populations visiting both commercial and educational sites.

The absolute values of traceroute divergence may not be completely indicative of the proximity of a client to its

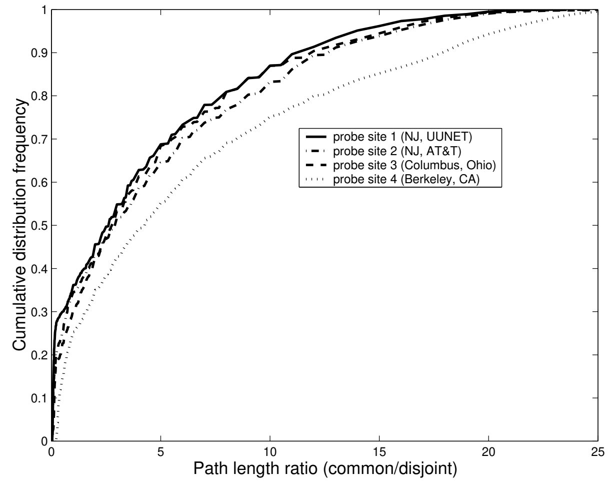


Figure 3: Ratio of common to disjoint path length

local DNS server. In Figure 3, we plot the ratio of the common path length to the disjoint path length from a probe site. Using the terminology of Shaikh, et al. [15], the common path length is the minimum number of network hops of the shared path from the probe site to the local DNS server and the client before their paths diverge. For example, the common path length of client 112.74.197.163 and its LDNS 112.25.195.1 (shown in Table 6) is $min(11, 11) = 11$. The disjoint path length is the maximum number of network hops of the diverging paths. In this example, the divergent path length is max(14-11, 13-11)=3. Again, we use the last point of the divergence as the reference point. For all probe sites, less than 34% of the client-LDNS pairs have disjoint paths at least as long as the common path. This means that at least 66% of client-LDNS pairs have a common path as long as or longer than their disjoint path. This metric implies that most clients are topologically close to their LDNS as viewed from a randomly chosen probe site.

### 3.3 Round-trip time correlation

Some CDNs select servers based on the round-trip latency between the CDN server and the client's local DNS server [15]. It is therefore important to understand the correlation between the round-trip delay to a client and to its LDNS from a third location.

To compare with the results presented in [15], we study how the round-trip delays to the client and its LDNS determine the accuracy of the CDN server selection based on round-trip delays to the LDNS. Since our data set consists of more than 4.2 million pairs of client and

LDNS, much larger than that presented in [15] (1,090 pairs), we expect some differences. Let $t_c^i$ and $t_d^i$ be the round-trip delays between the probe site $i$ and the client, and between the probe site $i$ and the client's LDNS, respectively. We ask the question whether $t_d^i < t_d^j$ implies $t_c^i < t_c^j$. Depending on the locations of two probe sites $i$ and $j$, the percentage of violations ranges from 17% to 38%. For instance, among the 9,360 client-LDNS pairs responding to traceroute from both probe site 1 and 2, about 38% violate this assumption. This implies that if one selects between two CDN servers located at probe sites 1 and 2 based on the round-trip delays to the LDNS, the decisions would be suboptimal 38% of the time for the set of clients considered based on the round-trip delay metric. On the other hand, among the 7,895 pairs responding to traceroute from both probe site 2 and 4, only 17% violate this assumption. This means that this metric is highly dependent on probe locations. However, it is a reasonable metric for use to avoid really distant servers.

Another interesting question to answer is whether, if two CDN servers are roughly an equal distance from the LDNS based on the round-trip delay, the same holds from the client's perspective. Thus, we ask whether $|t_d^i - t_d^j| \leq w$ implies $|t_c^i - t_c^j| \leq w$, where $w$ is a small number (e.g., a 10 ms threshold was used by Shaikh et al. [15]). In the sample of our study, it holds in 44–75% of the cases depending on the probe sites. This number is bigger than the previously obtained result of 12% in [15].

### 3.4 Improved local DNS configuration

For the client and local DNS associations that are not in the same network cluster, we ask whether there exist any local DNS servers in those clusters. From our log, we collected a set of local DNS servers. Thus, assuming the clients have access to those local DNS servers in their network clusters, it is interesting to examine the degree of improvement if all LDNS servers were used optimally. This assumption is not unreasonable, since most IP addresses in the same network cluster are under the same administrative control. From Table 4, we can calculate the number of client ASes and network clusters where there are no local DNS servers as observed in our log. There are $9,570 - 8,590 = 980$ such AS clusters, and $104,950 - 53,321 = 51,629$ such network clusters. Table 7 compares the improved percentages of client-LDNS associations and HTTP requests in the same cluster with the original results. If the clients in our data currently configured to use a LDNS in a different cluster are allowed to use an LDNS in the same cluster, then at least 92% of the HTTP requests come from clients using the LDNS in the same AS cluster. That number is 70% for network clusters.

Table 7: Improvement of the percentage of the client-LDNS associations sharing the same cluster using optimal LDNS assignment

| Metrics | Client IPs | | HTTP requests | |
|---|---|---|---|---|
| | Original | Improved | Original | Improved |
| AS cluster | 64% | 88% | 69% | 92% |
| Network cluster | 16% | 66% | 24% | 70% |

### 3.5 Clients using multiple local DNS servers

Some client IP addresses in our data are associated with multiple LDNS IP addresses. This may happen due to the following reasons: (1) The first LDNS server the client contacts times out and the second LDNS server is contacted. (2) The client's LDNS server is configured by a DHCP server that assigns the LDNS server IP addresses from a set of addresses in a round-robin fashion. (3) A client may be configured to round-robin among multiple LDNS servers. (4) The client IP address is reused at different times by different users and these users may have different configurations for their LDNS servers, resulting in different associations. (5) The client IP address is that of a NAT box or a application-level proxy, so there are multiple actual clients behind this IP address using different LDNS servers. (6) The client is misconfigured.

Here we examine the distribution of the LDNS servers with which a client IP address is associated. If they all occupy the same cluster as the client, DNS-based server selection can use the local DNS server's IP address to estimate where the client is even if the client uses multiple local DNS servers. However, if they occupy multiple clusters or a single cluster different from the client, it is more difficult to use DNS-based server selection. In Table 8, we show how many clients use ten or fewer local DNS servers. In addition, we observe that some IP addresses are associated with up to 330 local DNS servers occupying up to 273 different network clusters. Further investigation shows that some of these addresses belong to cache proxies. In general, we observe that the more LDNS servers with which a client IP address is associated, the lower the percentage of associations with the client and LDNS in the same cluster. Fortunately, the majority of client IP addresses are associated with a single LDNS server. They are responsible for about 52% of the requests. However, only about 20% in this group

Table 8: Clients using ten or fewer multiple local DNS servers

| # of clients (% of total) | # of LDNS (avg # of NACs) | % of total HTTP requests | % associations with client and LDNS in the same NAC |
|---|---|---|---|
| 2,524,939 (78.064) | 1 (1.0) | 51.8 | 20.3 |
| 522,228 (16.146) | 2 (1.6) | 22.4 | 12.1 |
| 123,524 (3.819) | 3 (2.1) | 10.4 | 6.6 |
| 41,422 (1.281) | 4 (2.5) | 4.9 | 4.7 |
| 13,469 (0.416) | 5 (2.9) | 2.5 | 4.9 |
| 4,555 (0.141) | 6 (3.3) | 1.8 | 6.7 |
| 1,590 (0.049) | 7 (4.1) | 1.3 | 9.9 |
| 713 (0.022) | 8 (4.7) | 0.7 | 13.6 |
| 461 (0.014) | 9 (5.5) | 0.7 | 14.2 |
| 273 (0.008) | 10 (6.1) | 0.5 | 14.0 |

Table 9: Comparison of four proximity metrics

| Proximity metric | Evaluation |
|---|---|
| AS clustering | 78% in the same cluster |
| Network clustering | 23% in the same cluster |
| Traceroute divergence (TD) (probe site 2) | 16%: TD=1, 32%: TD=2<br>median TD=4, mean TD =5.7<br>65%: $disjointPathLen \leq commonPathLen$ |
| RTT correlation (probe sites 2, 3) | 71%: $t_d^2 < t_d^3 \Rightarrow t_c^2 < t_c^3$<br>62%: $\lvert t_d^2 - t_d^3\rvert \leq 10ms \Rightarrow \lvert t_c^2 - t_c^3\rvert \leq 10ms$<br>$a = t_d^2 - t_d^3, b = t_c^2 - t_c^3$<br>$correl(a, b) = 0.13$ |

have the client and LDNS in the same network cluster.

### 3.6 Comparisons of proximity metrics

Given the above set of metrics for evaluating proximity between client and its local DNS server, we compare their results on a common set of 7,894[4] client-LDNS associations in Table 9. The comparison shows that network clustering is a fine-grained metric, similar to traceroute divergence (TD) count of 1. Hosts within the same network cluster, or which have a TD of 1, are guaranteed to be very close to each other. However, hosts not in the same network cluster, or have a TD bigger than 1, may still be quite close. Thus, these two metrics are quite conservative. AS clustering is the most coarse-grained metric, since an AS can be quite large. This is comparable to the ratio of common to disjoint path length. RTT correlation is also a relatively coarse-grained metric. It is inconclusive and largely dependent on the two probe site locations.

In general, performance-oriented metrics such as round-trip time should provide accurate real-time network latency measurements. CDNs often do real-time network measurements from their servers to clients. Since we can only probe from a limited set of locations, such metrics are inconclusive. Topology-oriented metrics have the advantage of being non-invasive, since they do not incur any network overhead. However, they cannot take network congestion into account.

As we explain in the following section, the applicability of each metric depends on the density of CDN server placement. The denser the placement, the more fine-grained metric is needed.

[4]Only 7,894 of all associations can be reached from both probe sites 2 and 3.

## 4 Application impact

In this section, we focus on the impact that client-LDNS associations have on DNS-based server selection. We study this impact in detail for three of the largest commercial CDNs. We anonymize the CDN names to properly reflect the nature of this work as a research vehicle rather than any form of competitive analysis. All three CDNs chosen rely on deploying caches in multiple networks. ISP-based CDNs deployed by companies like AT&T and Qwest are excluded from this study, since their caches are located in one or two ASes. Since a client and its LDNS are very likely to be in the same AS (about 69% of HTTP requests in our study), an ISP-based CDN can easily identify a peering link that is suitable for the AS containing both of them[5]. The results described below are representative of all the data we collected and remained stable during our entire study.

Previous work by Johnson, et al. [10] has shown that DNS-based CDNs do not always pick the best server available. Here we study whether this is partly due to the inherent limitations of DNS-based server selection. The answer to this largely depends on the proximity between clients and local DNS servers and the location of CDN servers.

The proximity evaluation of client-LDNS associations using the network clustering metric indicates that, if a CDN had a server in each network cluster, about 84% of the selection decisions for the client population in our log could be *suboptimal*. This is because our study

[5]The main tradeoff here is fewer peering links traversed in multi-ISP CDNs versus less traffic between access and backbone routers as well as lower costs in single-ISP CDNs.

found only 16% of these clients have their LDNS in the same network cluster. For clients with their LDNS in different network clusters, the CDN would most likely resolve the DNS query from a client's LDNS to the CDN server in the LDNS's cluster and not the cluster where the client resides. In reality, and as we show below, even the biggest CDN today does not have a CDN server in every network cluster. Thus, it is important to examine the impact of DNS-based redirection in a commercial content distribution setting.

We assume that on average a CDN server within the client's AS/network cluster or smaller traceroute divergence (TD) is closer than one in a different cluster or larger TD. For clients with CDN servers in their clusters, if a CDN selects a server not in a client's cluster, this may be a suboptimal decision in terms of proximity. We also assume that CDNs attempt to optimize for proximity in most cases. Network bandwidth is less important, since the content delivered by these CDNs is relatively small in size. Although CDNs may also incorporate the avoidance of overloaded servers in their server selection algorithms, we believe that our assumption is reasonable because CDNs today are highly overprovisioned from the perspective of server capacity. Furthermore, we repeated our experiments on separate dates to avoid any possibility of a skew due to a flash event, and the results were always similar. One limitation in our results below is that we do not quantify suboptimal server selection in terms of end user performance, nor how close it is to the optimal server selection.

We first describe our measurement methdology then use AS/network clustering and traceroute divergence to study how the proximity between client and LDNS affect DNS-based server selection in three commercial CDNs.

### 4.1 Experiment methodology

We use the following three data sets for our study.

1. **Client-LDNS associations.** These associations between clients and their LDNS servers are obtained from our measurement study.
2. **LDNS-CDN server associations.** For a given CDN, these associations map LDNS servers from the first data set to the CDN servers selected by the CDN when resolving a query from these LDNS servers.
3. **Available CDN servers.** This data set represents a list of CDN servers available in a given CDN.

In the first data set, we sampled 42,991 LDNS servers from our measurement study. We obtained the second data set by sending DNS queries to these 42,991 LDNS servers using the *dig* command for a domain name of a Web site that we know is a customer of a given CDN. 27,918 of these LDNS servers do not use access control and hence answered the queries from our machines, as if these machines were their clients. To answer our queries, these LDNSs recursively resolved our queries with the CDN in question. The server selected by the CDN for this DNS query is exactly the same server that would be used by any real client associated with this LDNS, as if that client and not our machine initiated the DNS query.[6]

The third data set was obtained in a similar way, except we added a large number of additional LDNS servers to the 27,918 LDNS servers above, for a total of 41,754 different local DNS servers. This is to increase the likelihood of finding all CDN servers of a particular CDN for a given domain. The extensive list of geographically distributed LDNS servers was obtained from DNS server logs for a large Web site. The set of servers to which a given CDN resolved queries from these LDNSs represents the servers available in this CDN at the time of the experiment. We obtained our second and third data sets at around the same time each day to find the set of servers available to a CDN at the time it performed its server selection in the second experiment.

Note that our set of available servers is conservative, since we might not have discovered all available CDN servers. However, if a CDN performs a suboptimal server selection among a subset of all available servers, its server selection will remain suboptimal for a larger set: suboptimal means that we already found a *closer* server to the client than the one selected by the CDN. A superset of the list of servers would suffer from the same suboptimal assignment.

Many CDNs claim a much larger number of caches. However, CDNs do not utilize all servers for all Web sites and many of their locations may contain multiple caches. The statistics we gathered are for a particular domain served by a CDN. For example, when examining multiple different domain names served by the largest CDN in our study, we found multiple CDN IP address sets of approximately equal size which only partly overlapped. Each unique server IP address we discover may also account for multiple servers.

[6]Note, for fault-tolerance, most CDN DNS servers usually return multiple IP addresses. In this case, we pick the first one, since clients also typically choose the first IP address.

Table 10: CDN cache servers for a particular domain name

| CDN | # of AS clusters with servers | # of network clusters with servers | # of CDN servers IPs |
|---|---|---|---|
| CDN X | 622 | 740 | 1,567 |
| CDN Y | 120 | 152 | 195 |
| CDN Z | 60 | 79 | 154 |

Table 11: The evaluation of server selection according to AS clustering

| CDN | CDN X | CDN Y | CDN Z |
|---|---|---|---|
| Clients w/ CDN server in cluster | 1,679,515 | 1,215,372 | 618,897 |
| Verifiable clients | 1,324,022 | 961,382 | 516,969 |
| Misdirected clients (% verifiable clients) (% clusters occupied) | 809,683 (60%) (92%) | 752,822 (77%) (94%) | 434,905 (82%) (94%) |
| MC w/ LDNS not in client's cluster (% misdirected clients) | 443,394 (55%) | 354,928 (47%) | 262,713 (60%) |

Table 10 shows the statistics of the CDN server IP addresses of the three CDNs studied for a single domain name obtained on August 7, 2001. These numbers were fairly stable during the course of our study. All three CDNs examined appear to redirect client requests by using DNS, although they may differ in the details of the algorithms. This table lists the total number of CDN servers discovered and the number of AS and network clusters these CDN servers represent. The data in Table 10 confirm our conjecture that CDNs today cover only a small number of all available network clusters for a single domain they serve. While the overall list of LDNSs used for generating the third data set represents 5,788 AS and 21,786 network clusters, the discovered CDN servers represent only a small fraction of these, even in the case of the largest CDN in our study.

With the three data sets above, we evaluate the quality of server selection by these CDNs by examining what percentage of clients are actually redirected to servers in their own cluster, among those clients that have at least one server in their cluster.

Table 12: The evaluation of server selection according to network clustering

| CDN | CDN X | CDN Y | CDN Z |
|---|---|---|---|
| Clients w/ CDN server in cluster | 264,743 | 156,507 | 103,448 |
| Verifiable clients | 221,440 | 132,567 | 90,264 |
| Misdirected clients (% verifiable clients) (% clusters occupied) | 154,198 (68%) (77%) | 125,449 (94%) (82%) | 87,486 (96%) (93%) |
| MC w/ LDNS not in client's cluster (% misdirected clients) | 145,276 (94%) | 116,073 (93%) | 84,737 (97%) |

## 4.2 Results of DNS-based server selection in commercial CDNs

Tables 11 and 12 show the results of our server selection evaluation using AS and network clustering. We collected 3,234,449 distinct client IP addresses in our logs. The first row of the table contains the number of clients with CDN servers in their clusters for the considered CDNs. Depending on the server density of each CDN, the number of clients with servers in their AS clusters ranges from 19% to 52% of the total clients in the log. This fraction is an order of magnitude lower in the context of network clusters. Thus, according to either metric, most clients will have to be served by *remote* servers. But a more interesting question is how many clients that could have been served by local servers are in reality directed to remote ones.

To answer this question, we concentrate on clients with servers in their clusters and consider the LDNS-CDN server associations for these clients from the second data set. Unfortunately, not all of these LDNS servers respond to DNS queries from our machines. The second row of the tables gives the number of clients, among those with CDN servers in their clusters, whose LDNS servers responded to our queries. We call these clients *verifiable* because we could find out which CDN servers a CDN would redirect these clients to. The third row shows the number of clients that a CDN directed to an external CDN server (one that was outside the client's cluster), when there was an available CDN server within that cluster. We refer to such clients as *misdirected* clients (MC) based on the assumption that CDN servers within the cluster are closer than external ones, although we accept that other factors than proximity may have affected the assignment. We see a large number of misdirected clients according to both proximity metrics. To confirm that these misdirected clients are not due to any anomaly of clients belonging to a small number of clusters, we also show in the third row the percentage of clusters occupied by these clients relative to the total number of clusters of verified clients. The cluster percentage values are at least as big as the client percentage values. This means that the misdirected clients are fairly spread out in the number of clusters they occupy.

We conjecture that the reason that these clients are misdirected is that their LDNS servers are topologically distant from these clients. CDNs select a server close to the LDNS servers. The servers selected may therefore be suboptimal from the client's perspective. The last row of the tables shows misdirected clients with their LDNS outside their clusters. This row indicates the number of clients that inherently cannot be directed to the most proximal server using a DNS-based mechanism. According to Table 11, for AS clustering, they represent only half of misdirected clients. To understand why CDNs choose a CDN server in a different AS than the one containing the client and its LDNS server, we sampled a dozen of these clients using *traceroute* followed by DNS name resolution of the last-hop router IP address to estimate the geographic locations[7] of the client, CDN servers in the client's AS, and selected CDN servers in a different AS. We found that in most cases, the selected CDN servers by CDNs are geographically closer to the client than CDN servers in the same AS. Assuming peering links between the client's AS and the selected CDN server's AS are not congested, redirecting to a nearby CDN server in a different AS may be a better decision than redirecting to a distant CDN server in the same AS. This observation also confirms our finding that AS clustering is a very coarse-grained metric for evaluating proximity.

For network clustering, the last row of Table 12 indicates that an overwhelmingly *majority* of misdirected clients have their LDNS servers in a different network cluster. This confirms our hypothesis that such misdirection is due to the fact that clients and their LDNS servers are often not proximal. It also shows the usefulness of network clustering because it is a fine-grained metric for evaluating proximity. We emphasize that we do not know the exact server selection policy used by a commercial CDN, so we cannot fully evaluate the effectiveness of its server selection decisions. However, given that there is such a strong correlation between misdirection and an LDNS being in a different cluster, we can infer that when the LDNS and client do not belong to the same network cluster, this limits the accuracy of server selection.

[7]In many cases, the router's DNS name has an indication of the geographic location [14].

Table 13: The evaluation of server selection according to traceroute divergence (TD) from probe site 3

| CDN | CDN X | CDN Z |
|---|---|---|
| Client-LDNS pairs examined | 2,105 | 2,171 |
| Clients with CDN servers at smaller TD than ones redirected to | 1,606 (76%) | 1,724 (79%) |
| Median TD of CDN servers clients redirected to | 11 | 13 |
| Median TD of closest CDN servers to clients | 5 | 9 |
| Median TD improvement | 6 | 4 |

Table 13 shows the evaluation of DNS-based server selections according to the traceroute divergence metric.[8] We performed traceroute from probe site 3 to a sample of client and local DNS servers from the log and the CDN cache servers from the third data set. The sample is chosen by randomly selecting one client-LDNS pair from the top 1200 client clusters generating the most HTTP requests. We found over 70% of the clients to be directed to a CDN server that is more distant than another available CDN server. Selecting the closest CDN server would have reduced traceroute divergence by as much as 19 hops for some clients.

Overall, we conclude that, among the clients we could verify, knowing the client's IP address would allow more accurate server selections in a large number of cases (443,394 for CDN X). The last row of Tables 11 and 12 also indicate the number of improved CDN server selections if the client's IP address were known to the CDN. Relative to the total number of clients, in the case of CDN X, this represents a small percentage: specifically 14% (443,394 out of 3,234,449). In general, the number of misdirected clients depends on the server density, placement, and selection algorithms.

## 5 Related work

Our work is motivated by a related effort by Shaikh, et al. [15] examining the effectiveness of DNS-based server selection. They developed a method of finding client-LDNS associations using time correlations of DNS and HTTP requests from DNS and Web server logs. However, as they have noted, the associations obtained using their method are inherently inaccurate due to clock skews, client DNS caching, and mishandling of TTLs. To resolve ambiguities, they used heuristics based

[8]We were unable to include CDN Y in the traceroute experiment, since most of its CDN servers are unreachable using traceroute.

on AS numbers and domain names to decide whether a client and a nameserver did in fact belong together. This heuristic removed misconfigured client-nameserver pairs and did not assure the correctness of associations. They also obtained a set of 1090 client-LDNS associations from accounts with 9 commercial ISPs to study the proximity correlations.

In comparison, our method provides accurate associations eliminating any need for validation. Furthermore, our study has more than 4.2 million associations, consisting of clients from a diverse set of ISPs, far exceeding their data set of 1090 associations.

More recently, Bestavros, et al. [4] have also developed a method for finding client-LDNS associations by assigning multiple IP addresses to a Web server and correlating DNS lookups with client IPs based on the server IP used. This method is slow in discovering client-LDNS pairs due to the limited number of IP addresses a Web server can have. In addition, their method is complicated to implement, requiring reassignment of server IPs and modification of the Web server.

Compared to both works, the distinguishing features of our measurement methodology are efficiency, nonintrusiveness, and accuracy. This allowed us to collect more extensive data, which we used to evaluate the effectiveness of DNS-based server selections using four different proximity metrics in several real-world CDN settings. To our knowledge, we are the first to conduct such an exhaustive proximity evaluation between clients and their local DNS servers using such a representative data set. We are also not aware of other work in examining the impact that the proximity between the local DNS server and the client has on DNS based server selection in commercial CDNs.

There has been a recent effort within the IETF to categorize different mechanisms for request routing in CDNs [3]. DNS-based redirection is one of those mechanisms, and our methodology may prove useful in evaluating the effectiveness of this technique in that context.

## 6 Future work

There are three areas of future work we plan to pursue. First, we plan to study the hidden load factors due to differing amounts of HTTP load corresponding to a DNS name resolution request from an LDNS server. With the help of a busy Web site, we will be able to gather statistics on the number of HTTP requests and clients behind each LDNS server. Identifying LDNS servers resulting in large numbers of HTTP requests is essential for proactive load balancing and flash crowd protection.

Second, we plan to improve existing DNS-based server selection algorithms by considering the properties of known client-LDNS associations for an LDNS that requests a server name resolution. The following characteristics of the associations can be explored based on data collected using our methodology: known client proximity to the LDNS, known client distribution, and hidden load factor.

Given a name resolution request from an LDNS, if the known client proximity to the LDNS is good, then a CDN server close to the LDNS would also be close to its clients. If the proximity correlation is low, known client distribution and client cluster request patterns would be considered. If the majority of HTTP requests belong to a single network cluster, finding a CDN server close to or within that network cluster would also be close to clients issuing a majority of requests. Along with these factors, the hidden load factor of the LDNS is also considered to select lightly loaded CDN servers for an LDNS with a large hidden load factor. If the proximity correlation is low between LDNS and its clients, then server selection is optimized using other metrics such as server load.

Finally, we would like to apply the results of this work to improving content distribution internetworking (CDI), which refers to the interoperation among multiple CDNs for additional flexibility. A prototype of CDI, called *CDN Brokering* [6], uses a DNS-based brokering mechanism to forward requests among DNS servers of the interoperating CDNs. As a third area of future work, we plan to improve CDN brokering algorithms by using hidden load factors and client-LDNS proximity information. The client-LDNS proximity findings in our work justify DNS-based brokering, because the majority of the clients and their LDNS belong to the same AS.

## 7 Conclusion

In this paper, we propose a novel technique for finding client and local DNS server associations and potentially hidden load factors in a fast, non-intrusive, and accurate manner. Based on the results, we evaluate the proximity between clients and their LDNS using four metrics: AS clustering, network clustering, traceroute divergence, and round-trip time correlation.

We evaluate the potential effectiveness of DNS-based server selection in CDNs based on these metrics. We conclude that DNS is good for very coarse-grained server selection, since 64% of the associations belong to the same AS. DNS is less useful for finer-grained server selection, since only 16% of clients use a DNS server in the same network-aware cluster. These values can be improved to 88% and 66% respectively, if clients are configured to use a closer local DNS server. Since current CDNs are not present in many network-aware clusters, we conclude that although DNS-based server selection has inherent limitations due to potentially poor proximity correlation between a client and its LDNS, the impact is small due to the sparse distribution of CDN servers in today's CDNs.

At least one CDN has stated a goal of ultimately placing CDN servers in every edge network. The high fraction of clients using LDNS servers in different network-aware clusters suggests that CDNs may be unable to use DNS request routing for such fine-grained server selection unless DNS itself scales to provide each edge network with a local DNS server that communicates directly with the Internet. Thus, from an economic perspective, due to the inherent limited precision of DNS-based server selection, it is less beneficial to have so many CDN servers that the performance to two nearby servers is indistinguishable.

In addition to the proximity evaluation and the novel measurement methodology, our work also provides two additional contributions in improving DNS-based CDNs in general. From our observation, client-LDNS associations are fairly static. Thus, CDNs can build up a database of such associations to infer the geographic location of clients associated with an LDNS IP address to improve server selection. Furthermore, based on the URL-rewriting technique in our measurement methodology, CDNs can completely eliminate the originator problem by embedding the client IP addresses in the URLs of the Web pages, when a client initially requests the base page.

## 8 Acknowledgement

We thank all participants in our measurement study. We especially thank Ted Kowalski of AT&T, Danielle Gallo of AT&T Research, Alex Brown and Milan Andric of UC Berkeley, Frans Kaashoek of MIT, and Mike Dahlin of UT Austin for generously offering their main Web pages for our study. We are grateful to Robert Szewczyk and Alec Woo for instrumenting the TinyOS Web page. We also thank Hyunseok Chang, Yuan Gao, Jason Hong, David Oppenheimer, Amit Sehgal, Wilson So, and Hao Zhang, who took part in our measurement using their personal Web pages.

## References

[1] Edge side includes. `http://www.esi.org`.

[2] Keynote systems. `http://www.keynote.com/`.

[3] A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck. Known CN Request-Routing Mechanisms, February 22 2002. Work in Progress, draft-ietf-cdi-known-request-routing-00.txt.

[4] Azer Bestavros and Sumit Mehrotra. DNS-based internet client clustering and characterization. Technical Report BUCS-TR-2001-012, Boston University, 2001.

[5] Telstra Internet. `http://www.telstra.net/ops/bgptab.txt`.

[6] Alex Biliris, Charles D. Cranor, Fred Douglis, Michael Rabinovich, Sandeep Sibal, Oliver Spatscheck, and Walter Sturm. CDN brokering. In *Proceedings of 6th International Workshop on Web Caching and Content Distribution*, June 2001.

[7] M. Colajanni, P. S. Yu, and D. M. Dias. Analysis of task assignment policies in scalable distributed Web-server systems. *IEEE Transactions on Parallel and Distributed Systems*, 9(6):585–600, 1998.

[8] Michele Colajanni, Philip S. Yu, and Valeria Cardellini. Dynamic load balancing in geographically distributed heterogeneous web servers. In *Proceedings of the 18th IEEE International Conference on Distributed Computing Systems (ICDCS98)*, May 1998.

[9] Geoff Huston. Internet bgp table. `http://www.telstra.net/ops/bgp/`, November 2001.

[10] Kirk L. Johnson, John F. Carr, Mark S. Day, and M. Frans Kaashoek. The measured performance of content distribution networks. In *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, 2000.

[11] B. Krishnamurthy, C. Wills, and Y. Zhang. On the use and performance of content distribution networks. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW'2001).*

[12] Balachander Krishnamurthy and Jia Wang. Topology modeling via cluster graphs. Proceedings of SIGCOMM IMW 2001.

[13] Balachander Krishnamurthy and Jia Wang. On Network-Aware Clustering of Web Clients. In *Proceedings of ACM SIGCOMM'2000*, 2000.

[14] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for internet hosts. In *Proceedings of the ACM SIGCOMM 2001*.

[15] Anees Shaikh, Renu Tewari, and Mukesh Agrawal. On the effectiveness of DNS-based server selection. In *Proceedings of IEEE Infocom 2001*, 2001.