Diversity in DNS Performance Measures

Richard Liston, Sridhar Srinivasan, Ellen Zegura

Abstract— **DNS is a critical component of the operation of Internet applications. However, DNS performance in the wide-area is not well understood. A number of studies present DNS performance measurements [1], [2], [3], [4], but the measurements are out of date, are not collected at client locations (e.g., they are taken at root servers), or are collected at very few client locations.**

In this paper we present the largest known study of wide-area DNS performance at clients. We use data obtained under a variety of network environments such as location in the Internet topology, connection technology and client ISP. We identify DNS system performance measures and investigate the degree to which they vary from site to site. We report on measures that are relatively consistent throughout the system such as the fraction of names whose lookups succeed, and those that vary widely, such as overall response times and response times from root and gTLD servers. We also discuss the impact of some of these measures on DNS performance for non-cached domain names, confirming some current notions about DNS operation while challenging others.

Keywords— **DNS, performance, active measurement.**

I. Introduction

The Domain Name System (DNS) is primarily used to resolve the human-readable domain names of remote machines to IP addresses. This resolution, called a domain name *lookup*, is typically the initial step in communication between two IP endpoints when the remote IP address is not known. Thus, DNS is a critical component of the operation of many Internet applications.

Descriptions of the operation of DNS are provided in several other papers [1], [2], [5], [6], [7], and the details of this process are available in the relevant RFCs [8], [9]. The aspect of DNS we wish to highlight is that clients in different locations may experience very different performance while looking up the same names. We illustrate some possible reasons for this difference in Section II.

Richard Liston, Sridhar Srinivasan and Ellen Zegura are with Georgia Institute of Technology, E-mail: {liston,sridhar,ewz}@cc.gatech.edu .

This work was supported by DARPA under contract number F30602-99-1-0514.

When evaluating Internet application performance it is often important to evaluate the impact of DNS performance. Indeed, there are studies and tools that evaluate the latencies contributed by DNS to web performance [4], [10], [11], [12], [13]. There are also some studies that specifically target DNS performance [1], [2], [3]. However, these studies are old, or consist of performance measurements taken from a very limited number of locations (1 or 2) in the Internet topology, or do not focus on performance from the perspective of the client. This raises the question "To what extent does DNS performance vary across Internet clients?" The answer has implications regarding the equity of Internet infrastructure services and the usefulness of DNS performance studies from a small number of vantage points.

Evaluating DNS performance at more than one location is difficult since the method of taking measurements, e.g., using `tcpdump`, may require privilege that is hard to obtain in many domains. But performance may, in fact, vary from location to location due to local differences in available bandwidth, network architecture and proximity to the elements supporting DNS resolutions such as root servers, generic top-level domain (gTLD) servers, as well as other servers like country-code top-level domain (ccTLD) servers and the authoritative servers for the domain names being resolved.

In this paper we present the largest known study of wide-area DNS performance at clients. We investigate the degree to which metrics for wide-area DNS performance, such as mean response time, number of servers contacted, and root and gTLD server performance, differ across locations in the Internet.

DNS employs caching to increase performance. A cached domain name record circumvents wide-area DNS operation, so lookups for this name are not subject to variations in the wide-area resolution mechanism. Some studies demonstrate that even when caching is enabled the lookup times for domain names can be quite long. Wills and Shang [10] report lookup times exceeding 2.0 seconds for as many as 29% of lookups to random servers, and Cohen and Kaplan [14] report lookup times exceeding 3.0 seconds for as many as 10% of lookups. They also note that "caching is applicable to only about 30% to 50% of requests." It is for non-cached names that users will expe-

rience the longest lookup times. For this reason we study the behavior of DNS only for non-cached domain names[1].

We seek to identify metrics that are invariant or vary little with respect to changes in factors like topological location, connection technology and ISP. We also identify metrics that are sensitive to changes in these factors, and we report the observed ranges for these metrics.

The metrics we investigate include the number of servers that must be contacted to resolve names. Each server that must be contacted adds a round-trip time that depends upon the proximity of each server, network conditions and server load. We quantify the percentage of lookups from our data set that complete, either finding the answer or finding that the name does not exist. We investigate the overall response times, as this metric reflects the user's experience of system performance. Metrics we explore that are expected to be invariant across locations are fraction of names that are aliases (they return CNAME, or canonical name, records), the TTLs (time-to-live) of returned lookups and the fraction of names that are successfully resolved. Finally, our analysis of root and gTLD server response times may be useful in guiding future engineering of the DNS system as well as other global distributed systems. Our key findings are as follows:

- There is a wide range in DNS performance for resolving non-cached names. The mean response time for completed lookups varies from 0.95 seconds to 2.31 seconds.
- Response time from gTLD servers has a very noticable impact on the mean time to resolve non-cached domain names; gTLD servers are queried during approximately 60% of the lookups at each site, and account for 13.9% to 28.9% of the mean response time.
- Response time from root servers has a negligible impact on the mean time to resolve non-cached domain names; root servers are queried during approximately 7.0% of the lookups at each site, and account for only 1.5% to 3.4% of the mean response time.
- The set of root and gTLD servers that provide the best service changes from site to site. There is more variation among gTLD servers that provide the best service than among root servers.
- The proportion of names that are aliases varies little across sites. A very small percentage of the names that are aliases receive different CNAME mappings.
- The distribution of TTLs of completed lookups is not sensitive to location.

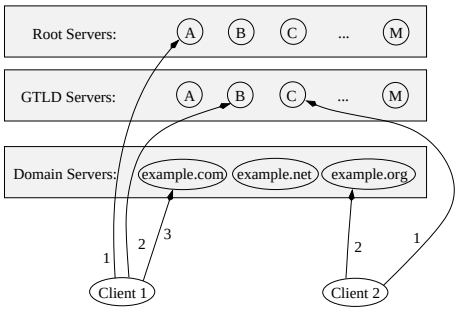

Fig. 1. Using root and gTLD servers to resolve the same name at different clients. The numbers indicate the order in which queries are sent.

II. Differences in DNS Performance

This section reviews DNS behavior and terminology, and also illustrates how two clients can experience very different performance during normal DNS operation while resolving the same name. In Figure 1 we have two clients[2] that are looking up the name foo.example.org. In this example, Client 1 has no information cached about foo.example.org, .example.org or .org. The first request is sent to the root server A.ROOT-SERVERS.NET. The response to this query indicates that the client must query one of the thirteen gTLD servers. This type of response is called a *referral*. Client 1 chooses B.GTLD-SERVERS.NET for this query[3]. To increase reliability, it is recommended that domain information be replicated at different servers, and that they "be placed at both topologically and geographically dispersed locations on the Internet" [17]. So in our example, the response is a referral to one of two servers that should have the answer: example.com and example.org. Client 1 chooses example.com, sends the query and receives the answer.

Client 2, on the other hand, has some information about .org cached, and sends a query directly to C.GTLD-SERVERS.NET, bypassing the initial root server query. Client 2 receives a referral to example.com and example.org also, but continues by sending a query to example.org and receives the answer. Each response contains a time-to-live (TTL) indicating how long the client may

[1] As described later, some aspects of the lookup may be cached (cf. Section II).

[2] In this study we make no distinction between a client and its DNS server. Although it has been shown that only 16% of DNS client/server assocations are in the same network-aware cluster [15], DNS performance at the client is strictly subject to the performance experienced at the server.

[3] From the web page describing djbdns [16]: "*dnscache* simply contacts a random server, to balance the load as effectively as possible. BIND keeps track of the round-trip times for its queries to each server, with various bonuses and penalties, and then sends all its queries to the 'best' server".

Fig. 2. Locations of root and gTLD servers.

cache the answer.

Figure 2, obtained from CAIDA [3] and updated to reflect two additional gTLD servers in Atlanta and Seattle, further illustrates the potential for clients to experience different DNS performance depending on location. The figure shows the locations "of the root nameservers and gTLD servers. The (x,y) notation near the city names indicates the number of root servers (x) followed by the number of gTLD servers (y) in that area. Notice the large number of both types of servers around Washington D.C. and in California." [3]. The map highlights the fact that the root and gTLD servers that are central to the operation of DNS are geographically concentrated in the U.S., with many geographic regions entirely unrepresented.

III. Measurement Methodology

We had three primary goals in developing our tool to collect DNS performance data. First, we wanted to capture fine-grained information about the operation of the DNS system. A low level of detail provides us with great flexibility in analyzing DNS performance. The difficulty is that much of the operation of the system is, by design, hidden from the client [8]. The local DNS server performs most of the work on behalf of the client, querying any servers that need to be queried until an answer is found, or until it cannot proceed any further. However, the interaction between the local DNS server and the rest of the system determines the performance from the user's point of view. Second, we wanted to collect data in such a way as to make comparison between sites as meaningful as possible. Limiting the kind of data collected at each site and controlling the method by which it is collected reduces the error in our comparisons. Finally, we wanted to be able to collect data at multiple locations. The more locations in which we collect data, the stronger our statements are about global DNS performance.

To meet these goals, we created a tool to run independently at multiple data collection sites to actively perform measurements. The primary component of the tool is the `named` name server[4], modified to log each event that advances the server towards resolution of the names, with a timestamp on each line in the log. We post-processed the logs for subsequent analysis. The events we logged were as follows: receipt of a request to resolve a name; the sending of a request to remote servers; the receipt of responses from remote servers; the answer sent to the querying client; the removal of queries from an internal queue of pending queries; the identification of an entry in the local cache; and the identification by the server of the type of the response. We then packaged the modified name server with a script to drive the name lookups (the client application), a list of names to be resolved, and configuration files that allowed the tool to be run by a non-privileged user at a specific port. The script utilizes the `dig` command, which invokes the resolver library `gethostbyname()` function. This causes the client script to issue a request directly to the modified server at the server's port and to wait for a response. After 5 seconds if an answer has not yet been received the resolver times out and repeats the request. The resolver returns immediately after the second request has been sent, but the server continues attempting to resolve the name via retries for tens of sec-

[4] `named`, `dig` and `gethostbyname()` are provided with the Berkeley Internet Name Domain (BIND) software distribution.

Seed page	Type of web site	Host organization type	Country
www.cnn.com	News	Commercial	US
www.gatech.edu	Information	Educational	US
www.parismatch.tm.fr	Entertainment	Commercial	France
www.house.gov	Information	Government	US
www.chimfunshi.org.za	Information	Non-Profit	South Africa
www.sina.com.cn	News	Commercial	China
hptdc.nic.in	Information	Government	India
8ball.federated.com	Entertainment	Individual	US

TABLE I
STARTING POINTS FOR CRAWLER COLLECTING DOMAIN NAMES.

onds. For this reason events for different name lookups may be interleaved in the logs, requiring special care during log processing.

A. Measurement Locations

The data was collected in three groups: on NIMI [18] nodes, by colleagues with accounts on remote machines, and by members of the Linux user community who were willing to participate in this study[5]. We obtained measurements from 75 different Internet locations in 21 countries and territories[6]. Data from an additional seven sites were discarded due to anomalies in the collection process. For example, at some sites the connection was broken for some interval of time during the collection period because the individual had exceeded their maximum login time.

While we did not systematically collect specific information about the kind of Internet connection at every site where measurements were taken, many of the participants freely offered this information. From this we know that the measurement locations represent a wide variety of connection technologies, including DSL, PPP, cable modem, gigabit ethernet, etc. The timestamps in the logs show that data was collected on different days of the week, and at various times of day at each site. The dates of collection fell into two primary periods: January 2002 and late March/early April 2002.

It is unlikely that participating clients interfered with each others' measurements. Queries from our server to remote servers do not (as is normal) have the recursion desired flag set, so the remote servers should not retrieve the result in order to cache it. In Section III-C we argue that the increased load on the system has a minimal affect on performance for other participants.

[5]We emailed requests directly to individuals who identified themselves as contact persons for Linux User Groups worldwide.

[6]Countries and territories represented in our data set are: Argentina, Australia, Brazil, Costa Rica, Czech Republic, Denmark, France, Germany, Greece, Italy, Japan, Northern Ireland, Norway, Poland, Russia, Slovak Republic, South Korea, Spain, Sweden, Switzerland, United States.

B. Domain Name Sample

To perform measurements on DNS performance we first collected a large number of domain names (around 100,000). The names were collected by crawling the Web with the *Larbin* crawler [19]. Given a seed page, this crawler fetches and parses it, then recursively follows links on the resulting pages. To branch to as many web sites as possible while minimizing the impact on network and server performance, we used the default configuration of following links five deep into a web site with 60 seconds between successive requests to the same server, and reduced the number of parallel connections from 200 to 10.

The set of web sites reached by crawling the Web is highly sensitive to the starting point for the crawler [20], so we crawled multiple times, each time seeding the crawler with a different starting point drawn from a set of pages that differ in several parameters. The set of pages we chose to seed each crawl represented different values of variables such as popularity, type of web site, country of origin and type of hosting organization. The pages we used, along with the characteristics we chose to vary, are listed in Table I.

Many of the names we collected at this stage were not valid domain names. This appears to be a consequence of many factors such as mistyped links and incorrect HTML syntax. We first removed all ill-formed names. We then restricted the names to valid top-level domains. However, some invalid, yet well-formed, names remained in the name sample. Since our goal was to measure DNS performance for non-cached names, we selected a set of names that were unique up to the second level. For example, if there were two names from the example.com domain such

TLD category	Percentage of names in category
com	50%
org	14%
net	9%
edu	6%
de	3%
ru	2%
fr	1%
ca	1%
gov	1%
it	1%
151 others	less than 1% each

TABLE II
TOP LEVEL DOMAINS IN THE DOMAIN NAME SAMPLE.

as a.b.example.com and c.example.com, we selected only one of these names for inclusion in the final set of sample names. We did not, however, want to completely remove the effects of caching, since the normal operation of a DNS server typically has useful information cached even when the target name itself is not cached. Such information may be gTLD or ccTLD servers, or remote servers that are "closer" to the domain name being resolved. Thus, we did not force the server to flush the cache after each name lookup.

The final data set consisted of 14,983 names, each representing a unique second-level domain. The resulting names fell into the top-level domain categories in the percentages shown in Table II.

C. Network Impact

When using active probing to perform measurements, we must quantify the impact on the system in terms of resource usage. Typically the collection takes about 4-6 hours of continuous operation to complete on each client. It requires roughly 40K outgoing packets of 40 bytes each and roughly 40K incoming packets of 300 bytes each, spread out over the entire run. On average this consumes about 700 bps outgoing and 5Kbps incoming at the measurement site.

An upper bound on the worst-case increase in root server load is calculated as the ratio X/Y, where X is the maximum number of queries across all clients to any single root server during a collection run and Y is the minimum amount of time across all clients for any collection period in seconds. The resulting worst-case increase in root server load is 832 packets/15641 seconds, or .053 packets/sec. Similarly, an upper bound on the worst-case increase in gTLD server load is 8868 packets/15641 seconds, or .57 packets/sec. These upper bounds are low enough to make certain that our measuments neither place a burden on the DNS system, nor perturb our measurements, even when multiple measurements are being taken simultaneously.

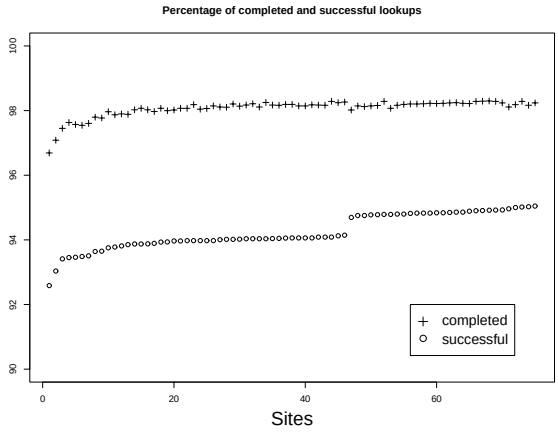

Fig. 3. Number of completed and successful lookups.

IV. RESULTS

In this section we examine several metrics and analyze the degree of variation in these metrics observed across the 75 measurement sites where our data collection tool was run. The primary metrics we investigate are the completion and success rates of resolving names; the mean response time for completed lookups; the root and gTLD servers that are favored by the sites; the observed fraction of names that are aliases; and the distribution of TTLs across names.

A. Completion and Success Rates

We first examine the rate of completion and the rate of success of each participating site. The response codes sent by our modified server to the client script consisted of the following values: 0, indicating no error occurred; 2, indicating a remote server failure; and 3, indicating the name does not exist. We consider a resolution to be *complete* if our server returned an answer with a response code of either 0 or 3. We consider a resolution to be *successful* if our server returned an answer with a response code of 0.

Figure 3 plots, for each site, the percentage of lookups that completed and the number that were successful. The range of values for completed lookups is [14500,14700], or 96.4% to 98.1% of the lookups. The number of successful lookups is in the range [13900,14200], or 92.7% to 94.7% of the lookups. This high number of successes can be attributed to the method by which we obtained domain names, and the filtering of invalid addresses. Ap-

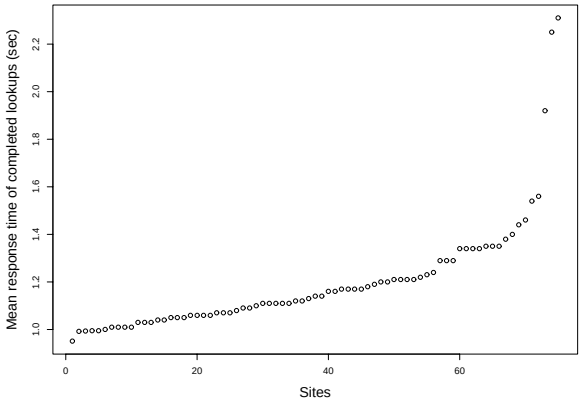

Fig. 4. Mean response times for completed lookups at each site.

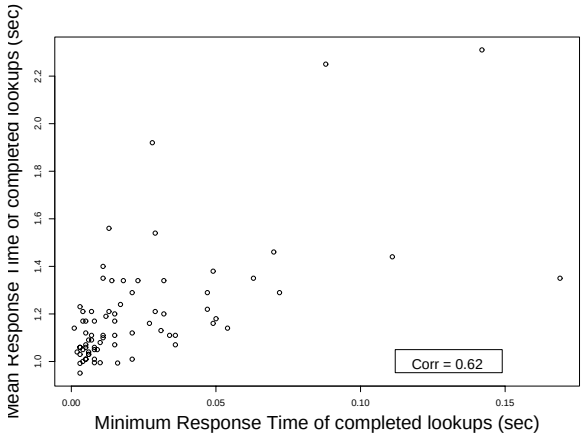

Fig. 5. Minimum response times vs Mean response times for completed lookups at each site.

proximately 3% of the lookups did not complete. This can be caused by factors such as unavailable nameservers, incorrectly configured nameservers and a lack of a route to nameservers. We do not have sufficient information to quantify these problems.

In Figure 3 the sites are ordered by the number of successful lookups, exposing an interesting phenomenon. There are two weak clusters: around 14,100 successful lookups for sites 3 through 46, and around 14,200 successful lookups for sites 47 through 75. Two sites have slighly lower numbers of completed and successful lookups and do not belong to either cluster. Examining the logs for these two sites, we noted that they experienced higher numbers of retries during portions of the data collection, lasting from 3 to 16 minutes. This is likely caused by the presence of congestion close to the collection point, causing the slightly lower number of completed lookups. Other logs also seem to have experienced short periods of congestion that caused higher numbers of retries, but the congestion was not so severe as to cause more lookups to fail.

Examining the sites for each of the clusters we note that they are grouped according to the dates of data collection. The data with the slightly higher number of successful lookups were all collected in January 2002 and those with a lower number of successful lookups were all collected in late March/early April 2002, showing that roughly 0.6% of the names became invalid over the course of about two months. We conclude that the numbers of completed and successful lookups for a static set of names are time-sensitive and, to the degree that one site experiences congestion more than another, they may also be location-sensitive.

B. Mean Response Time

Figure 4 shows the mean response times for completed lookups (MRTc) at each site, with sites ordered by MRTc. We see a large disparity in overall performance among the sites. The minimum MRTc is 0.95 seconds and the maximum MRTc is 2.31 seconds. This is a difference of 1.36 seconds, or a factor of 2.4. This is a very noticable delay for applications such as web browsing that require DNS lookups during human interaction.

We speculate that the four major factors affecting the MRTc for a site are the site's *connectivity*, *loss rate*, *perceived performance of root and gTLD servers*, and *location in the network relative to other name servers*. We do not have fine-grained information about each of these factors for many of our sites, so we devise methods to estimate each factor from our data, and then test for correlation with the MRTc to quantify the effect of the factor on the MRTc.

B.1 Connectivity

A site's connectivity is determined by the combination of its bandwidth and its proximity to the Internet. To investigate the affects of connectivity on MRTc, we assume that the Minimum Response Time for completed lookups (MINc) is a good measure of a site's connectivity. This quantity captures the minimum round trip time for a DNS query/response to the closest name server to which a query is made. A lower MINc should correspond to a higher bandwidth connection and/or close proximity to the Internet.

In Figure 5 we plot the MRTc against the MINc at each site. Our expectation was that those sites with the highest MRTc would also have the highest MINc due to poor connectivity. We do see in the figure that the two sites that

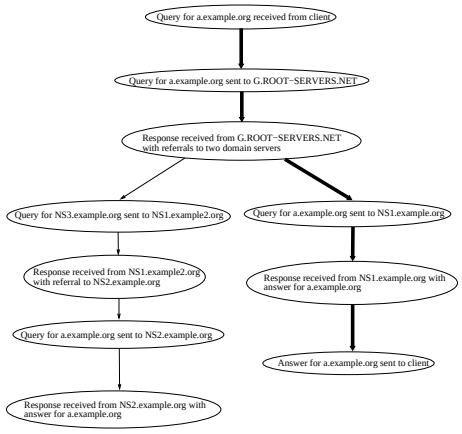

Fig. 6. Example resolution tree and its critical path.

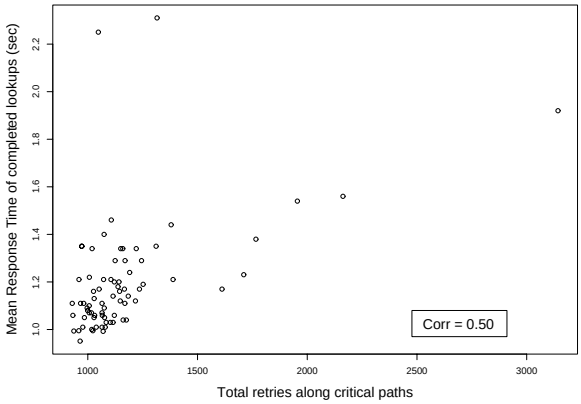

Fig. 7. Total retries along critical path vs MRTc at each site.

have the highest MRTc's (above 2.2 sec) also have higher MINc's. However, we also see two other sites that have high MINc's (above .10 sec) that also have significantly lower MRTc's (below 1.5 sec). We calculate the coefficient of correlation, ρ, of the MRTc and the MINc. As there is only a moderate correlation ($\rho = 0.62$) between the two variables, we conclude that connectivity does not sufficiently account for the higher MRTc's.

B.2 Loss Rate

The local DNS server often receives responses from remote servers that contain multiple NS (nameserver) records, or referrals, indicating other nameservers that should be contacted to resolve the name. It is quite common for the server to query multiple nameservers in parallel, leading to a "resolution tree", where each node in the tree represents a query or response sent between machines, and each directed edge between nodes represents a causal relationship between two nodes. For example, a response from a root server may cause a query to a gTLD server. The root of the tree represents the original request, and one or more leaves may contain the A (answer) record. One such resolution tree is illustrated in Figure 6.

We use the critical path analysis technique [21] to examine the loss rate[7]. The unique path from the root of the resolution tree to the first answer sent to the client (there are sometimes multiple answers sent) comprises the *critical path* of the lookup. In practice, we determined the critical path by identifying the first answer and following the edges in the reverse direction up to the root. The nodes and edges traversed comprise the critical path for the lookup. In Figure 6, the dark arrows indicate the critical path for this resolution tree. The local server maintains a timer for outstanding queries. When a query to a remote server, or its response, are lost the timer expires and the local server resends the query to the remote server. These *retries* may also occur on the critical path for a lookup.

In Figure 7 we plot the MRTc of each site against its total number of retries along the critical path. The correlation between critical path retries and MRTc is weak ($\rho = .50$). Under the assumption that retries are a good measure of loss rate, we conclude that loss rate is not a major factor affecting lookup time for our data set. We note, however, that the loss rate varies dramatically across sites.

B.3 Root/gTLD Server Performance

We analyze the impact of the performance of root and gTLD servers by calculating the Mean Response Time for all queries sent to root servers (MRTr). Similarly we calculate the Mean Response Time for all queries sent to gTLD servers (MRTg).

Figures 8 and 9 show plots of each site's MRTc against the site's MRTr and MRTg, respectively. Here we see a strong correlation for each ($\rho = 0.86$ and $\rho = 0.94$), initially suggesting that the performance of the root and gTLD servers have a major effect in the overall DNS performance at a site. We also see a broad range of MRTr and MRTg across the measurement sites. The range of MRTr is from 0.063 seconds to 1.41 seconds and the range of MRTg is from 0.037 seconds to 0.89 seconds.

To investigate the impact of root, gTLD and other server performance, we calculated the percentage of lookups where each of these server types was queried at some point along the critical path. The results are shown in Figure 10. We see that the percentages are quite constant at approximately 7.0% for root servers, 60.0% for gTLD servers and 98.4% for other servers. The seemingly high percent-

[7] The idea of using critical path analysis for a portion of our work was inspired by work by Barford and Crovella [22]. They used critical path analysis to investigate various effects on the critical path profiles of TCP transactions.

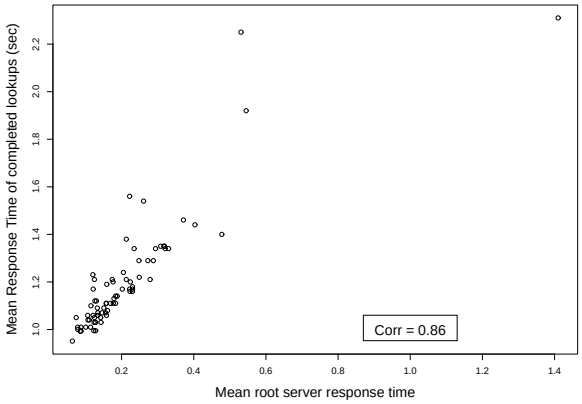

Fig. 8. Mean root server response time vs MRTc at each site.

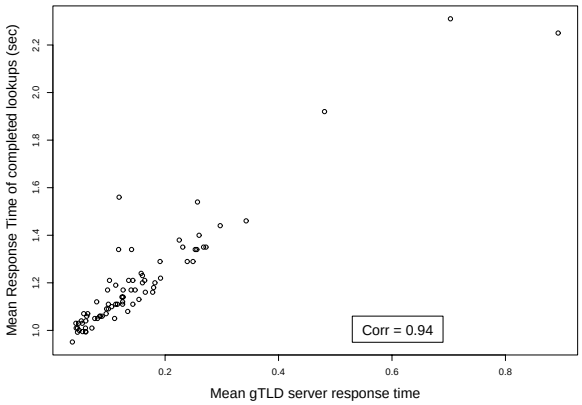

Fig. 9. Mean gTLD server response time vs MRTc at each site.

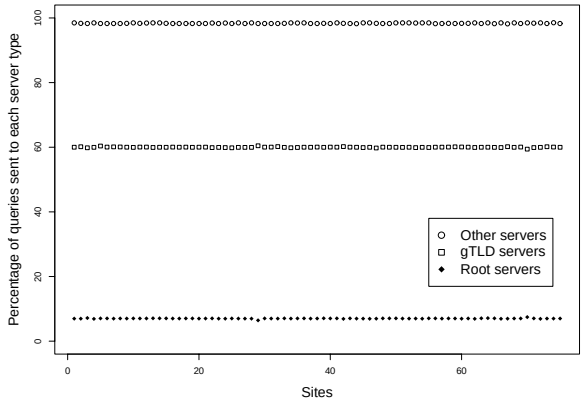

Fig. 10. Percentages of queries to root, gTLD and other servers.

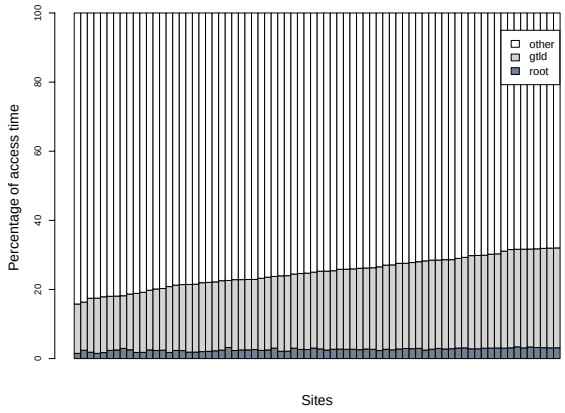

Fig. 11. Percentages of time querying root, gTLD and other servers along critical path.

age of queries that involve root servers is caused by the fact that the root servers delegate some domains at the top level (e.g., the .se domain), and some domains at the second level (e.g., the census.gov domain). So after the initial query for the .se domain, subsequent queries for this domain will circumvent the root server, but different second-level names under the .gov domain must still go directly to the root for referral. The percentage of time spent in the critical path querying each of root, gTLD and other servers is shown in Figure 11.

Some implications of these results are:

- The performance of servers other than root and gTLD servers have the largest impact on performance of lookups for non-cached names. Thus, schemes that reduce this portion of the lookups, such as those employed by content delivery networks (CDNs), have the greatest impact on speeding up lookups.
- For some sites, root and gTLD server performance is quite poor, taking as much as 1.41 seconds and 0.89 seconds on average, respectively, to respond to requests. Since about 60.0% of lookups involved gTLD servers but only about 7.0% involved root servers, poor performance is not as egregious for root servers as it is for gTLD servers.
- Also because of the difference in impact on performance by root and gTLD servers, a possible service differentiator delivered by an ISP is the performance it provides from gTLD servers for non-cached names. Far less important is the performance it provides from root servers. This could influence decisions regarding peering points and routing.

B.4 Network location relative to other servers

To estimate the location of a site relative to the rest of the Internet, we calculate the response times observed by each site to a fixed set of servers. This is related to the idea of distributed binning [23] where clients fix their location in the network based on measurements to a fixed set of servers.

We chose as our fixed set of servers the last server queried along the critical path while resolving names. We identified the set of servers that used the same set of IP addresses across all sites. We then extracted the names of the

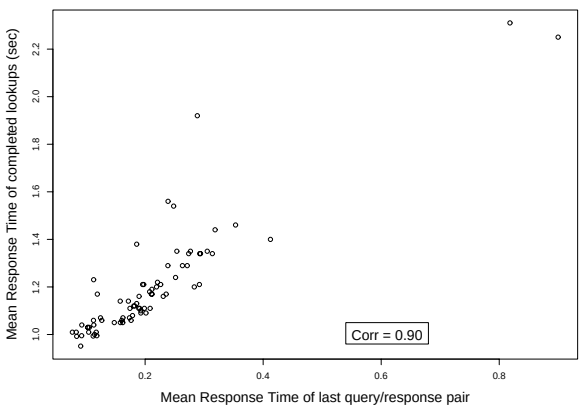

Fig. 12. Mean response time for last server vs MRTc at each site.

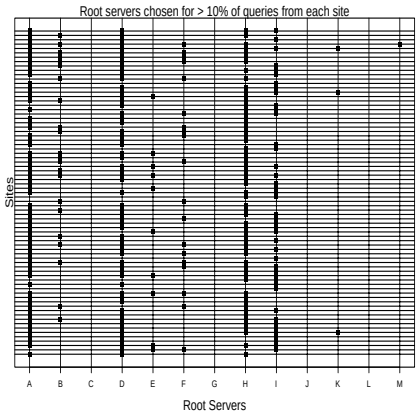

Fig. 13. Root servers favored by each site.

498 servers in this set and calculated the Mean Response Time from these last servers (MRTl) for each site. In Figure 12 we plot the MRTl against the Mean Response Time for completed queries (MRTc).

The method and results of this section are similar to those of Section IV-B.3 where we examined the correlation between root and gTLD server performance and the MRTc for each site. However, the size of those sets of servers is considerably smaller (13 of each instead of 498). From those results we only make conclusions regarding performance to those types of servers. With the larger set of servers used in this section, we can make more broad conclusions regarding distance to the rest of the Internet.

We find that the correlation between the MRTl and MRTc is strong ($\rho = 0.90$). This, under our assumption of response to the fixed set of servers indicating distance, demonstrates that the location of the site relative to the rest of the Internet is an important factor in the lookup time.

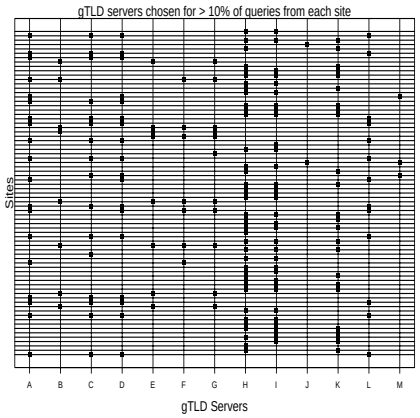

Fig. 14. gTLD servers favored by each site.

C. Root Server Interactions

The results for the root and gTLD servers in the previous section prompted us to further explore the interactions between local DNS servers and root and gTLD servers. BIND employs a server selection algorithm that seeks to minimize resolution times. The algorithm maintains a history of response times from servers when they respond to queries about a portion of the namespace. It ages this information so that all servers that will respond to queries for a portion of the namespace get sampled over time. In this section we consider the degree to which queries are distributed to the root servers from each measurement site, and the response times from the root servers.

We tallied the total number of responses (not just on the critical path) received from each of the root servers by each site. We then calculated the percentage of the total from each site. We say that a site *favors* a root server if it sends greater than 10% of its root queries to that root server. Figure 13 illustrates which root servers are favored by each site. Each vertical line represents a root server and each horizontal line represents a measurement site. A dark square placed at the intersection of a site line and root server line indicates that the site favored that server. We see that four root servers ($\{$A,D,H,I$\}$.ROOT-SERVERS.NET in Herndon, VA, US; College Park, MD, US; Aberdeen, MD, US; and Stockholm, SE, respectively) are favored by many of the sites, whereas six root servers ($\{$C,G,J,K,L,M$\}$.ROOT-SERVERS.NET in Herndon, VA, US; Vienna, VA, US; Herndon, VA, US; London, UK; Marina del Rey, CA, US; and Keio, JP) are favored by few or none of the sites. This does not indicate that that there are not sites that would favor these servers — only that the sites where we performed our measurements did not favor these servers.

D. gTLD Servers Interaction

Figure 14 has the same form as Figure 13, but for the gTLD servers. We see that two gTLD servers ({H,I}.GTLD-SERVERS.NET in Amsterdam, NL and Stockholm,SE) are favored by many of the sites, and two gTLD servers ({J,M}.GTLD-SERVERS.NET in Tokyo, JP and Hong Kong, CN) are favored by few of the sites. Again, this does not indicate that that there are no sites that would favor these servers - only that few of the sites where we performed our measurements favored these servers.

Comparing Figure 13 and Figure 14, we see higher preferences shown for fewer root servers than we see with gTLD servers. Favoring is distributed much more evenly among the gTLD servers. The result is that there is more variation in which gTLD servers are favored from site to site than for root servers.

E. Aliases and CNAMEs

A response from a nameserver of type CNAME indicates that the query was for a domain name that is an alias for a canonical name. It is possible for the resulting canonical name to be an alias for yet another canonical name, creating a chain of aliases. Whether a name is an alias or a canonical name is configured by the domain's administrator, and is not expected to vary as a function of location. However, some CDNs leverage the function provided by CNAMEs in DNS to increase performance of web object retrieval, so we expect some portion of the actual CNAME mappings to change from site to site.

Approximately 3960, or 26%, of the names in our data set were aliases. This percentage varied only slightly across sites as expected, and this slight variation may be attributed to the variation in the number of completed lookups. Table III shows the mean number (percentage) of CNAME aliases in the data set. The longest chain of aliases was 4. Only 51 of the 75 measurement sites saw the chain of 4 aliases.

We also investigated the variety of CNAMEs given for aliases while performing the name lookups. Table IV shows that by far most of the aliases resolved to the same canonical name (93.6%). Some aliases resolved to 2 and 3 different canonical names (5.9% and 0.2%, respectively), and 4 aliases resolved to 10 or more different canonical names, depending on site.

We conclude that, as expected, the number of names that are aliases is not location-sensitive, and that only a small portion of the actual CNAME mappings are location-sensitive.

Number of redirections, X	Mean number (percentage) of CNAMEs with X redirections
1	3810 (96.3%)
2	138 (3.5%)
3	8.77 (0.2%)
4	1 (0.03%)

TABLE III
CNAME REDIRECTIONS.

Number of different CNAME mappings, X	Number of aliases with X different mappings
1	4230 (93.6%)
2	269 (5.9%)
3	13 (0.2%)
10	1
11	1
15	1
19	1

TABLE IV
NUMBER OF DIFFERENT CNAMES PER ALIAS.

F. TTLs of completed queries

Because TTLs are set by the administrator of a domain and they should be a static value for each lookup, we expect each site to show the same distribution of TTLs for the answers. We investigated this by extracting the TTLs of the names that successfully completed. We then chose bins in which to count the number of TTLs. The bins were chosen somewhat arbitrarily based on the modes of the distribution of the TTLs for one site. We then calculated the number of TTLs in each bin across all sites. From each bin we took the maximum number and the minimum number, and took the difference as the range of values in each bin. To demonstrate the degree of the difference across the sites, we first calculated the range across all sites of the number of items in each bin. We then calculated the percentage of the mean represented by this range. The result is shown in Figure 15. On the x axis is each bin. (The chosen bins are shown more precisely in Table V.) On the y axis is the range as a percentage of the mean. We see that even with the variations in the number of names that were successfully completed at each site, the variation in the range of TTLs in each bin is extremely small. We conclude that, as expected, the distribution of TTLs seen at a

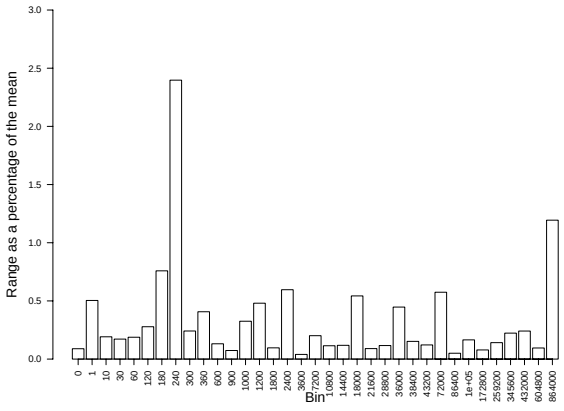

Fig. 15. Ranges of number of TTLs in each bin across all sites, as a percentage of the number of TTLs in the bin.

site is quite constant.

Readers may be interested in the distribution of TTLs across the names. We tabulate the mean number of items in each bin in Table V. The five most popular TTLs fall in the following bins, in decreasing order of popularity: [72001,86400], [2401,3600], [3601,7200], [38401,43200],[100001,172800].

V. Related Work

Studies focusing solely on DNS performance include [1] [2] [3]. The most widely-quoted study of DNS was published in 1992 by Danzig et al. [1]. Their study focused on measurements taken at one DNS root server and three domain servers. They identified a number of errors in DNS implementation and estimated the number of packets due to each type of error, and argue that the benefits of negative caching are outweighed by the benefits of a correctly administered system of nameservers.

Jung et al. [2] conclude that while it is widely believed that the two primary contributing factors to DNS performance are hierarchical design around administratively delegated namespaces and the aggressive use of caching, caching of NS records and avoiding overloading any single name server are more important factors. Their study is limited, however, in that they measure DNS interactions at only two points in the Internet topology.

Brownlee et al. [3] present measurements at a root server (F.ROOT-SERVERS.NET) and highlight problems that occur at clients that have adverse effects on root servers. They show that many problems observed by Danzig persist, and suggest that negative caching would help with the repeated query bugs.

Each of the three above studies is based on "live" workload sets at one or two locations, while our study constructs a single workload set to drive data collection in

Bin end value (sec)	Mean number in bin
0	236
1	24
10	52
30	64
(one minute) 60	128
120	32
180	11
240	15
(five minutes) 300	398
360	22
600	281
900	380
1000	18
1200	108
1800	331
2400	23
(one hour) 3600	2664
7200	562
10800	331
14400	414
18000	33
21600	399
28800	369
36000	83
38400	125
43200	526
72000	144
(one day) 86400	5524
100000	54
172800	407
259200	134
345600	72
432000	37
604800	115
(ten days) 864000	18
>864000	23

TABLE V
Mean number of names in each TTL bin.

multiple locations. Although our methodology does not use real user workloads, thereby prohibiting conclusions about user-perceived performance, it allows for strong comparisons of results from multiple locations.

Wills and Shang [10] investigate the effect of DNS lookup times on Web latency. Their three-part study involved replaying cache logs to study cache usage, measuring response time using DNS resolver routines, and measuring end-to-end performance of web page retrieval including DNS lookup time. This study is also performed at one location.

Cohen and Kaplan [14], [24] propose and evaluate several approaches for reducing the latency of Web transfers, with particular emphasis on DNS-related techniques like DNS prefetching and cache renewal policies.

Cranor et al. [7] present a method for characterizing the types of remote entities sending DNS requests. Using traces at backbone routers along with some other external data sets, they identify clients as DNS clients, local DNS servers, authoritative DNS servers and outliers.

Our results demonstrate that studies involving certain DNS performance measures would be greatly strengthened by data from many locations, or by considering the range of DNS performance that sites may experience.

VI. Conclusions

In this paper we have presented a fine-grained study of the operation of the DNS system from multiple locations in the Internet. Our goal was to compare various measures from different locations to determine which measures vary based on location and the degree to which they vary, and which measures remain relatively constant. This information will both help to guide engineering of the DNS and other global distributed systems as well as guide future studies that rely on DNS performance information. We examined the correlation of DNS performance, as determined by the mean response time for completed queries, with various metrics.

Our results have demonstrated that, as expected, the DNS system tends to have low variation in those measures that are controlled by site adminstrators, like the fraction of names that are aliases and the distribution of TTLs across those names. Other measures tend to vary widely as a function of location. These include mean response time for completed queries and response time from root and gTLD servers.

We show that the greatest performance enhancements can be achieved by reducing the response time of servers other than root and gTLD servers. We also show that reducing the response time of gTLD servers, possibly via more equitable choice of placement of the servers, has the potential to have a very noticable impact on perceived performance. In addition, we demonstrate that root server performance has a negligible effect on perceived performance. For those measures that vary widely as a function of location, we have demonstrated that measurements from few locations may not represent the range of performance experienced across the Internet.

VII. Future Work

The following is a list of some questions (including a few suggested by the anonymous reviewers of this paper) that may be pursued in the future. Many of these questions can be explored using our current data set, and others will need additional data for analysis. Our data is available to the public[8] for independent validation of our results, for investigation of the following questions and for assistance in other analyses. We encourage those who have accounts on machines in countries not yet represented in our data set (there are many) to run our collection tool [25] on those machines and forward the results to us for inclusion in future studies.

- How often do requests to root (and gTLD) servers fail due to timing out?
- Are there correlations between particular domains and number of servers contacted?
- Are there any common sources of misconfiguration with respect to domain delegation? How often do they occur?
- What is the typical distribution of the response times at a site?
- How many servers are typically contacted in order to load a single web page? What fraction of name lookups during a user's browsing session are for non-cached names?
- How much extra traffic is generated by following up on outstanding queries when the original query has already completed, either by failure or success?
- Is there an optimum timeout for retries by a resolver? How many queries would benefit from moving the timeout to the optimal time?
- To what degree are nameservers replicated? To what degree does the replication increase success rates? Would further replication help?
- What methods and guidelines should be used to ensure that a representative sample of Internet performance has been obtained?
- In practice, what are the DNS performance gains provided by CDNs that leverage DNS?
- How much does the DNS performance vary across ISPs in the same location?

[8] `http://www.cc.gatech.edu/computing/Telecomm/dnsperf/`

VIII. Acknowledgements

We thank Vern Paxson and Andrew Adams for their invaluable assistance in collecting data on many NIMI nodes; the members of the Linux community worldwide who contributed their time, and in some cases paid for the connection time, to collect data in countries where it would have been very difficult to obtain otherwise; Matt Sanders for creating the packaging that simplified the collection of the data; and Joe Hooper for preliminary investigation of how to best modify `named` to collect the desired data. Finally, we thank the anonymous reviewers for their valuable suggestions.

References

[1] Peter B. Danzig, Katia Obraczka, and Anant Kumar, "An analysis of wide-area name server traffic: A study of the domain name system," in *Proceedings of ACM SIGCOMM*, January 1992.

[2] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "DNS performance and the effectiveness of caching," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, 2001.

[3] Nevil Brownlee, Kimberly Claffy, and Evi Nemeth, "DNS measurements at a root server," in *Global Internet 2001*, November 2001, Presentation slides are available at http://www.caida.org/outreach/presentations/ietf0112/dns.damage.html.

[4] Girish Chandranmenon and George Varghese, "Reducing web latency using reference point caching," in *Proceedings of IEEE Infocom*, April 2001.

[5] Matthias Grossglauser and Balachander Krishnamurthy, "Looking for science in the art of network measurement," in *IWDC*, 2001, pp. 524–535.

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

[7] Charles D. Cranor, Emden Gansner, Balachander Krishnamurthy, and Oliver Spatscheck, "Characterizing large DNS traces using graphs," in *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[8] P. V. Mockapetris, "RFC 1034: Domain names — concepts and facilities," Nov. 1987.

[9] P. V. Mockapetris, "RFC 1035: Domain names — implementation and specification," Nov. 1987.

[10] C. Wills and H. Shang, "The contribution of DNS lookup costs to web object retrieval," Tech. Rep. TR-00-12, Worcester Polytechnic Institute, July 2000.

[11] Md Ahsan Habib and Marc Abrams, "Analysis of sources of latency in downloading web pages," Tech. Rep. TR-99-4, Virginia Tech, July 1999.

[12] Christian Huitema, "Internet quality of service assessment," http://www.netsizer.com/daily/quality_today.html.

[13] Christian Huitema and Sam Weerahandi, "Internet measurements: the rising tide and the DNS snag," in *ITC Specialist Seminar, IP Traffic Measurement, Modeling and Management*, 2000.

[14] E. Cohen and H. Kaplan, "Prefetching the means for document transfer: A new approach for reducing web latency," in *Proceedings of IEEE Infocom*, 2000.

[15] Zhuoqing Morley Mao, Charles D. Cranor, Fred Douglis, Michael Rabinovich, Oliver Spatscheck, and Jia Wang, "A precise and efficient evaluation of the proximity between web clients and their local DNS servers," in *Proceedings of USENIX 2002*, June 2002.

[16] D. J. Bernstein, "djbdns," http://cr.yp.to/djbdns/notes.html.

[17] R. Elz, R. Bush, S. Bradner, and M. Patton, "RFC 2182: Selection and operation of secondary DNS servers," July 1997.

[18] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "Creating a scalable architecture for internet measurement," *IEEE Communications*, vol. 36, no. 8, pp. 48–54, August 1998.

[19] Sebastien Ailleret, "Larbin," http://larbin.sourceforge.net.

[20] Balachander Krishnamurthy and Jennifer Rexford, *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*, Addison Wesley, 2001.

[21] K.G. Lockyer, *An Introduction to Critical Path Analysis*, New York:Pitman Publishing Company, 1964.

[22] Paul Barford and Mark Crovella, "Critical path analysis of TCP transactions," in *Proceedings of ACM SIGCOMM*, 2000, pp. 127–138.

[23] Sylvia Ratnasamy, Mark Handley, Richard Karp, and Scott Shenker, "Topologically-aware overlay construction and server selection," in *Proceedings of IEEE Infocom*, June 2002.

[24] E. Cohen and H. Kaplan, "Proactive caching of DNS records: Addressing a performance bottleneck," in *Proceedings of the Symposium on Applications and the Internet (SAINT)*, January 2001.

[25] Richard Liston, "Worldwide DNS Performance Study," http://www.cc.gatech.edu/~liston/dnsperf.html.

[26] S. Ratnasamy and J.M. Gonzalez, "A trace-based study of domain name system (DNS). implementation issues, usage patterns, and performance," May 2000.

[27] David Meyer, "Route views," http://www.antc.uoregon.edu/route-views/.