Availability, Usage, and Deployment Characteristics of the Domain Name System*

Jeffrey Pang
Carnegie Mellon University
jeffpang@cs.cmu.edu

James Hendricks
Carnegie Mellon University
jimi@cs.cmu.edu

Aditya Akella
Carnegie Mellon University
aditya@cs.cmu.edu

Roberto De Prisco†
University of Salerno
robdep@unisa.it

Bruce Maggs†
Carnegie Mellon University
bmm@cs.cmu.edu

Srinivasan Seshan
Carnegie Mellon University
srini@cmu.edu

ABSTRACT

The Domain Name System (DNS) is a critical part of the Internet's infrastructure, and is one of the few examples of a robust, highly-scalable, and operational distributed system. Although a few studies have been devoted to characterizing its properties, such as its workload and the stability of the top-level servers, many key components of DNS have not yet been examined. Based on large-scale measurements taken from servers in a large content distribution network, we present a detailed study of key characteristics of the DNS infrastructure, such as load distribution, availability, and deployment patterns of DNS servers. Our analysis includes both local DNS servers and servers in the authoritative hierarchy. We find that (1) the vast majority of users use a small fraction of deployed name servers, (2) the availability of most name servers is high, and (3) there exists a larger degree of diversity in local DNS server deployment and usage than for authoritative servers. Furthermore, we use our DNS measurements to draw conclusions about federated infrastructures in general. We evaluate and discuss the impact of federated deployment models on future systems, such as Distributed Hash Tables.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer-Communication Networks; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Measurement, Reliability

Keywords

DNS, federated, availability

1. INTRODUCTION

The Domain Name System (DNS) is an important part of the Internet's infrastructure and has grown to be one of the largest distributed systems in existence. DNS is also fairly complex, consisting of many different components ranging from the root and top-level nameservers to authoritative servers, local name servers, and client resolvers. Despite its complexity and importance to the functionality of nearly all network applications today, in-depth studies of its characteristics have only been carried out in the past few years. However, no attempt has yet been made to characterize the properties of the DNS *infrastructure* itself: How available is it? How are servers deployed? And which servers are used the most?

In this paper we present a large scale measurement study of the DNS infrastructure currently deployed in the Internet, focusing on characteristics that are indicative of how DNS servers are deployed and managed. In particular, we analyze availability and load characteristics of DNS servers, as well as deployment styles within different organizations.

Understanding these properties is important not only for the study of DNS, but also for the study of *federated* infrastructures in general — that is, the study of infrastructures that are managed independently by multiple organizations. For example, the DNS infrastructure gives us insight into how well infrastructure deployed primarily by publishers (authoritative name servers) is managed, versus infrastructure deployed by organizations primarily for the benefit of their users (local nameservers). Several recent studies [18, 28, 35] propose wide-area infrastructures to provide scalable, automated, and fault-tolerant lookup services, and some advocate or suggest deployment in a federated manner. Like DNS, future global-scale Internet infrastructures may be managed by a multitude of large and small organizations, rather than by a single entity. Hence, it is important to understand the characteristics of current federated deployment styles.

Our results indicate that a large fraction of all end-users use a small number of local DNS servers. We find that the majority of local and authoritative DNS servers are highly available, without a

*This work was supported by funding from IBM, the US ARO (via the C3S center – grant number DAAD19-02-1-0389), and the NSF (via ITR Awards ANI – grant number 033153, and CCR – grant number 0205523).

†Roberto De Prisco and Bruce Maggs are also affiliated with Akamai Technologies.

IMC'04, October 25–27, 2004, Taormina, Sicily, Italy.

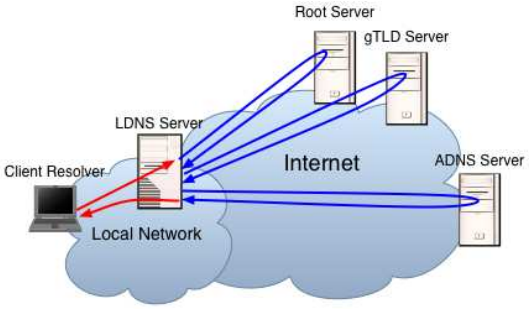


Figure 1: DNS Infrastructure Components: DNS infrastructure is comprises 3 types of components: client resolvers, local DNS (LDNS) servers, and authoritative DNS (ADNS) servers, of which the Root and Top Level Domain (gTLD) servers are special cases. To perform a lookup, client resolvers make a request to their configured LDNS server, which iteratively queries the ADNS servers required to resolve the requested name-to-address mapping.

single observed failure. The remainder have consistent availability periods lasting days, weeks, or longer, and recovery rarely requires more than a few hours. However there exists a fraction with only "one 9" (or less) of availability. In addition, we find that there is a slight positive correlation between usage and availability for both local and authoritative servers. Authoritative servers tend to have higher availability than local servers, especially if they have high usage, though the difference is not substantial. We also observe that there is a larger degree of diversity in local DNS server deployment and usage, with many behind dynamic IP addresses and some with diurnal availability patterns. Finally, we observe that local DNS server deployments within organizations range from a very few highly used servers to a very large number of lightly loaded ones, suggesting different levels of partitioning of administrative domains within different organizations.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of the different types of DNS servers and the infrastructure deployment styles that they exemplify. Section 3 describes our data collection methodology, our measurement testbed, and the data sets used in our analysis. Section 4 describes our measurement results and analyzes the load characteristics, availability, and deployment styles of DNS servers. To demonstrate the utility of our measurements, Section 5 presents an evaluation of a Distributed Hash Table (DHT) having DNS-like availability characteristics. Section 6 lists previous and related work. Section 7 concludes.

2. BACKGROUND

In this section we provide an overview of the different components that make up the DNS infrastructure and explain how their characteristics may be representative of other federated infrastructure deployments. In addition, we define the characteristics we measure and explain their significance.

2.1 DNS Infrastructure

The Domain Name System is a distributed database responsible for translating *names* to *addresses* of network locations, and its functionality is a critical component to almost all network applications including web browsing and email. Despite its simplicity of purpose, the implementation of DNS comprises many different components.

Figure 1 shows an example of the interaction between the various components [25] in DNS during a name lookup. A lookup is made by a *client resolver*, a software library implementing the DNS protocol on an end-user machine. These requests are forwarded to a *local* DNS (LDNS) server, which is usually deployed by the client's organization (for example, its university department or ISP). LDNS servers are important to DNS scalability because they cache DNS records. LDNS servers iteratively query the authoritative DNS servers (described next) to satisfy lookup requests on behalf of their clients. First the root servers are queried, then the gTLD servers, and then one or more additional authoritative DNS servers (though caching of DNS records usually makes the first few queries unnecessary).

The DNS database itself is distributed across the hierarchy of *authoritative* DNS (ADNS) servers. At the top of this hierarchy are the root and top-level domain nameservers. These servers do not generally maintain name-to-address records for Internet hosts (`A` records), but instead maintain records for locating ADNS servers at lower levels of the hierarchy (`NS` records), which contain the actual name-to-address records. ADNS servers at lower levels are deployed by a litany of organizations and store name-to-address records for resources associated with a particular organization (e.g., a university or company's web site).

Hence, LDNS and ADNS servers are two examples of federated service deployments. However, the motivations for maintaining each class of DNS server differ. LDNS servers are typically deployed within an organization to serve its own user population, while ADNS servers are typically deployed by publishers (or by enterprises that service publishers) in order to advertise a service or location to potential clients. One question we seek to answer is whether these different motivations and constraints on server deployment are correlated with observable characteristics, such as availability.

Answering this and related questions about availability, usage, and deployment characteristics may be important for future federated services, since such services could have deployment and management models similar to DNS. Indeed, several recent proposals [9, 28] have suggested moving DNS itself to a more decentralized lookup architecture due to the administrative problems and vulnerabilities associated with the existing hierarchical model. Other Internet naming layers have also been proposed [4, 35] to supplement or replace it.

2.2 Infrastructure Characteristics

We seek to understand three particular attributes about DNS infrastructure, listed below:

1. *Load Distribution*: The amount of load on each nameserver, which we define as the number of queries that it receives, is indicative of the number of users that access it.

2. *Availability*: The availability of a server determines how often it will be able to service requests. We define availability as the fraction of time during which the server machine is online when its local network can be reached. Both these conditions are usually necessary in order for the service to be available; nameservers and other services often require connectivity to regions of the network other than their own.

3. *Deployment Styles*: A final area of interest in this study is the style in which individual organizations deploy sets of nameservers. For example, we are interested in whether an organization partitions its users equally among several local name servers, or if there is more diversity in the load among different nameservers within the same organization. These styles

indicate the degree of locality that might exist in federated deployments and the number of servers that may be located within each administrative domain.

We next describe how we infer these characteristics from currently deployed DNS servers. We then explain the salient observations we made about these characteristics for DNS. Finally, we evaluate and discuss how our observations can be used in the development of future distributed systems.

3. MEASUREMENT SETUP

To infer the characteristics of interest, we collected and correlated several data sets concerning DNS servers. To infer relative load characteristics about DNS servers, we examined DNS access logs from Akamai, a large content distribution network (CDN), and correlated HTTP request logs from a set of Web caches over a period of a week. To infer availability, we performed active probes to a large number of servers over a two week period. Finally, to study deployment patterns, we examined the relative load characteristics of servers located within the same organization.

The remainder of this section describes how we obtained our DNS server sample sets and relative load estimates, the methodology behind our active measurement study of availability, and several steps we used to filter and reduce the data we collected to eliminate outliers and observed anomalies.

Figure 2 summarizes the data sets we used in our measurement results and analysis.

3.1 Sample Sets and Load Estimation

Here we describe our collection of logs to obtain samples of LDNS and ADNS servers and estimations of their relative loads.

3.1.1 LDNS Servers

The sample of LDNS servers we use in our relative load estimations and active probing measurements were obtained through the collection of DNS logs from Akamai's "low-level" servers, which should be close to the LDNS servers that access them. For each DNS record requested, the logs contain the time of access, the IP of the requester, and the name and type of the DNS record accessed.

Akamai operates the authoritative DNS for parts of a large number of popular websites (in particular, the parts of websites for which Akamai also hosts the contents, such as images), and we expect that the volume of requests a single LDNS server makes to this Akamai service is a good relative estimator of the actual load or client population behind it. For example, in a set of HTTP logs collected from a number of Web caches (described in Section 3.1.2), we observed that 14% of all HTTP requests made went to this Akamai service, and this Akamai service served the DNS for 26 of the top 100 most popular web site domains. Moreover, the DNS records returned by Akamai to LDNS servers have a TTL of only 20 seconds, so it is more likely that we will see a DNS request for each HTTP request than if examining ADNS logs with longer TTL records (though TTLs are not always respected).

Due to constraints in our log collection infrastructures, we could only collect DNS logs for approximately one eighth of the 8,500 active servers in Akamai. We ensured that we collected logs from all the servers in a particular *region* (e.g., ISP X in Chicago, IL) so that we would observe all the requests coming from LDNS servers in that region (Akamai's top-level DNS servers try to direct LDNS servers to the low-level server that yields the best performance, which is typically also close geographically). This portion of Akamai included regions from 49 different countries; 60% of the regions were in the U.S, and many were in Europe and Southeast Asia, in addition to other regions. Hence, our sample included LDNS servers deployed around the entire world. Moreover, the relative load distribution of LDNS servers observed in this sample set is nearly identical to the relative load distribution across LDNS servers seen by a larger (60%) fraction of Akamai servers during a shorter time interval, so we believe that any bias introduced by only having observed this subset of regions is unlikely to be significant.

We collected two sample sets over a one-week period from March 17 to March 24, 2004:

`ldns-all`: This data set includes all 823,694 distinct LDNS IPs seen in the logs collected from the subset of Akamai servers. We define the relative load on these LDNS servers as number of valid A-requests they made to Akamai during the week, calling this data set `ldns-load`. We only examined A-requests because those are the only records that properly behaving LDNS servers access from

Data Set	Sec.	Description
DNS Server Sample Sets		
`ldns-all`	3.1.1	All the LDNS servers that accessed the portion of Akamai for which we collected logs during the period from March 17 to March 24, 2004. (823,694 distinct IPs)
`ldns-probed`	3.1.1	A set of LDNS servers that were used when gathering the `ldns-avail` measurements. These servers were obtained by randomly sampling LDNS arrivals to Akamai servers during the period from March 17 to March 24, 2004. (273,541 distinct IPs)
`ldns-nondynamic`	3.3.2	A conservatively chosen subset of `ldns-probed`, which we determined were unlikely to be using dynamic IP addresses, and we use for the majority of our availability analysis. (164,040 distinct IPs)
`adns-all`	3.1.2	All the ADNS servers that we observed during a "reverse-crawl" of the `.in-addr.arpa` authoritative hierarchy during December 2003. (87,111 distinct IPs)
`adns-probed`	3.1.2	The subset of `adns-all` that mapped to a valid IP address and were still responsive to DNS requests in the middle of April 2004. (68,155 distinct IPs)
`adns-web`	3.1.2	All the ADNS servers that were authoritative for at least one website (ignoring those that were served by Akamai) in logs collected from several Web caches collected from May 3, 2004 to May 9, 2004. (85,719 distinct IPs)
DNS Server Measurements		
`ldns-load`	3.1.1	Relative load on `ldns-all`, estimated based on the number of requests each made to Akamai during the 1-week period from March 17 to March 24, 2004.
`adns-load`	3.1.2	Relative load on `adns-web`, estimated based on the number of requests made to websites they are authoritative for, as seen from several Web caches during a 1-week period from May 3, 2004 to May 9, 2004.
`ldns-avail`	3.2	A set of exponentially (mean 1 hour) spaced DNS and/or ICMP ping probes to each server in `ldns-probed` made during a three week period (two weeks of probes for each LDNS site) from March 18 to April 11, 2004. Section 3.3 describes steps taken to filter this data.
`adns-avail`	3.2	A set of exponentially (mean 1 hour) spaced DNS and/or ICMP ping probes to each server in `adns-probed` made during a two week period from April 15 to May 1, 2004. Section 3.3 describes steps taken to filter this data.

Figure 2: Measurement Data Set Summary: This table presents a summary of the sample sets of DNS servers we used for each of our measurement experiments (top) and the measurement data sets we collected (bottom).

Akamai's low-level servers (higher level DNS servers in Akamai serve NS records). Erroneous A-requests constituted less than 1% of the DNS logs .

ldns-probed: This data set includes all the LDNS IPs we included in our active probing experiment, described in Section 3.2. To collect this sample, we continually downloaded the most recent 2 minutes of a log on a random server in Akamai (not restricted to the regions for which we collected full logs). We then immediately (within 3 minutes of the LDNS having made the DNS request to Akamai) probed the server with a DNS request for A.ROOT-SERVERS.NET and an ICMP ping (retrying each request 3 times over 30 seconds). We did this to ensure that we only tracked LDNS servers that were responsive to our probes. We collected 374,334 distinct IPs in this manner, of which 273,541 (74%) were responsive to DNS or ping, and we tracked them in our active measurement study described in Section 3.2.

We find it interesting that only 35% responded both to DNS and ping, while 21% responded only to ping and 20% responded only to DNS. We did not notice a difference in the relative load distribution among the servers that were responsive and those that were not (when looking at the LDNS servers that we also had load information for in ldns-load). However, we note that our sampling method is biased in favor of LDNS servers that make a large number of requests. In Section 4.2.2, we find that there exists a slight positive correlation between relative load and availability, so we may slightly overestimate the fraction of samples with higher availability when we use our samples to draw conclusions about availability properties.

3.1.2 ADNS Servers

We obtained two ADNS sample sets: adns-all and adns-web.

To obtain adns-all, we performed a "reverse-crawl" of the .in-addr.arpa domain, which reverse maps IP addresses to domain names. We use a methodology similar to the ISC Internet Survey [16]: first, we look up the nameservers responsible for each of {0,...,255}.in-addr.arpa. For each domain in which we discover an ADNS, we recursively look up its "children" (e.g., to recurse on 128.in-addr.arpa, we examined each of {0,...,255}.128.in-addr.arpa), etc.

We note that we only performed a single sequence of lookups for a given domain, so if we did not receive a response, we missed the entire address block that made up its subtree (we received successful lookups for 2,711,632 domains). In addition, some domains in .in-addr.arpa may not map to an ADNS server, even though there exists one responsible for that domain (perhaps due to configuration errors). Hence, our sample is probably an underestimate of the number of ADNS servers in operation, though it is sufficient for the purposes of our study.

We found 87,111 distinct IPs this way, of which 68,155 were responsive to DNS (and possibly ping) in mid-April 2004. We used these 68,155 in our active probing measurements (see Section 3.2). We call this smaller set adns-probed.

To get adns-web, we obtained one week of Web cache logs (May 3 to May 9, 2004) from the NLANR IRCache project [15][1], which maintains a set of top-level Squid caches serving various regions in the U.S. From this data set, we obtained a set of 85,719 ADNS servers responsible for the websites accessed by clients in the trace. 20,086 of these servers were also in adns-probed. We estimated the relative load on these servers by summing the number of HTTP requests in the Web cache trace made to websites for which each ADNS server is responsible.[2] Although the actual load will be impacted by factors such as the TTL of the DNS records returned to clients, we hypothesize that this gives us an estimate good enough to make the correlations in our analysis (e.g., between load and availability). We call this data set adns-load.

3.2 Active Availability Measurements

After obtaining the ldns-probed and adns-probed samples described above, we began active probes to measure their availability characteristics. For LDNS servers, we began the measurements immediately after we verified that they were responsive.[3] For ADNS servers, we began all measurements at the same time. We tracked each DNS server for approximately two weeks, as follows.

During the experiment we sent DNS and ICMP ping probes to each DNS server we tracked, with an exponentially distributed interval between probes, with a mean of 1 hour. Probes were originated from a well-connected machine at Carnegie Mellon University and were made by a custom event-driven application for efficiency. DNS probes to both LDNS and ADNS servers consisted of a query for the A record of A.ROOT-SERVERS.NET, and each probe was tried 3 times over a period of 30 seconds before we marked it as failed. We probed a given DNS server with whatever queries to which it was originally responsive (e.g., if it originally did not respond to ping we only used the DNS query). Although we could also have used TCP RST packets to track servers, due to the volume of servers we planned to track, we decided to use less invasive methods that were unlikely to trigger firewall alarms, etc.

We use exponentially distributed sampling intervals for reasons explained by Paxson [27]. Such sampling is unbiased because it samples all instantaneous signal values with equal probability. In addition, it obeys the "PASTA" principal, which says that the proportion of our measurements that observe a given state is (asymptotically) equal to the amount of time spent in that state [36]. To verify that our sample granularity was fine enough to apply this property to our limited measurement interval, we performed active probes to a random subset of 900 LDNS and 900 ADNS servers at a much higher fixed-rate of 1 probe per 5 minutes for 3 days and obtained nearly identical availability distributions to those obtained from our experiments with larger intervals.[4]

The most significant drawback in our measurement setup is that we are only observing the DNS servers from a single vantage point, so our probes will observe network failures along the path from our probe machine to the target server site as well. However, we take steps to reduce the impact of these, as described in the following section. There is trade-off between logistical difficulty and sample size (which is proportional to resource expenditure) when setting up a measurement study at multiple sites (e.g., synchronization, administrative policies, maintenance); we opted for a much larger sample size in order to observe the peculiarities of DNS servers.

We call the data set for LDNS and ADNS servers obtained from these experiments ldns-avail and adns-avail respectively.

3.3 Data Reduction

We took several steps to filter the data to reduce the impact of network failures on our measurements. In addition, we reduced

[1]The NLANR IRCache project is supported by National Science Foundation (grants NCR-9616602 and NCR-9521745), and the National Laboratory for Applied Network Research.

[2]If there were multiple ADNS servers authoritative for a website, we split the load evenly between the servers.

[3]This allowed us to estimate the impact of network failures on our measurements by correlating probes with accesses at Akamai, as described at the end of Section 3.3.

[4]These measurements were performed between July 27 and July 30, 2004.

our LDNS sample set because of the presence of dynamic IPs addresses.

3.3.1 Network Failures

First, we tried to identify periods during which network failures occurred close to our probing site or in the "middle" of the network, hence affecting a large number of measurements. To do this, we examined our aggregate probing log (containing the probe results to all sites combined) and removed all probes that fell within a 30-second window in which the fraction of failed probes exceeded 3 standard deviations of the failure rate of the preceding hour. This removed periods where a larger than normal number of probes experience correlated failures, which could indicate significant network failures in the middle of the network or close to our probing site. This removed 5.1 hours of data from our LDNS experiment (of 560 total) and 7.8 hours from our ADNS experiment (of 388 total), the longest period of which was about 1.5 hours, during which CMU had a confirmed network outage.

Next, we clustered IPs according to autonomous systems (ASs) and network aware clusters (NACs) [19]. A NAC is the set of IP addresses sharing the longest common routing prefix advertised via BGP (which are derived using BGP routing table snapshots from several vantage points, such as RouteViews [2]). Hence, the IP addresses in a NAC are likely to have access links in common and are likely to be operated under the same administrative domain.

We examined our combined traces for ASs with a particularly high rate of correlated failures by looking for *related* probe pairs — that is, closely spaced (within 2 minutes) probe pairs to servers within the *same* AS but within *different* NACs.[5] We proceeded as follows: First, we counted the number of related probe pairs that observed at least one failure — the *total* number of failure pairs. Second, we counted the number of related probe pairs that were both failures — the number of *correlated* pairs. The ratio of correlated failure pairs to total failure pairs gave us an estimate of the number of correlated failures in a particular AS. In the LDNS sample set we found that 70 ASes were responsible for abnormally high ratios (above 0.3), so we eliminated the 800 sites in these ASes. In the majority of the other ASes (in which we observed closely spaced failures), there were 0 correlated failures.

We note that we did not expect these steps to completely eliminate network failure observations in our data, but merely to limit their impact on our analysis. To estimate how well our filtering heuristics worked, we checked if any of the LDNS servers generated requests to Akamai within 1 minute of an observed failure during the one week for which log collection and active probing overlapped. Limiting ourselves to the 6,000 servers that generated the most requests to Akamai (since each of these servers had an average request rate to Akamai greater than 1.5 per minute), we found that about 15% of the failed probes during this period were within 1 minute of an observed request from the corresponding LDNS. Hence, if we excluded all network outages from our definition of availability, then our measurements would underestimate the actual availability of nameservers. Nonetheless, we believe our measurements would still be correct to within an order of magnitude. In our analysis of this data, which might still be sensitive to network failure observations (such as in Section 4.3), we take further precautions.

3.3.2 Dynamic IP Addresses

For the sample of LDNS servers, we performed one final filtering step before proceeding to measurement analysis. We discovered that in a fair number of cases, the LDNS servers we sampled used dynamic IP addresses (e.g., DHCP). Bhagwan *et al.* [5] found that the aliasing effects introduced by dynamic IPs result in significant underestimates in availability when measuring peer-to-peer clients, and we have observed the same effect with LDNS servers. In particular, when examining a database of 300,878 class C address blocks known to be dynamic IP pools obtained from a spam Realtime Black Hole List (RBL) [1], we found that 17,163 (6%) LDNS servers in `ldns-probed` were classified as dynamic. Moreover, 27,237 of the domain names obtained by reverse mapping LDNS IPs in `ldns-probed` contained a string like `dsl` or `dialup` (in the host portion), suggesting that they are behind DSL, cable, or dial-up links (though this does not necessarily imply they are dynamic). Because identifying dynamic IP addresses is difficult and to our knowledge there is no passive technique with any reasonable degree of accuracy, we choose to be conservative and only analyze LDNS servers that we were reasonably confident were not using dynamic IPs. We used the following three heuristics to do this classification. We keep an LDNS IP if it satisfied at least one of the heuristics:

1. If the domain name obtained by reverse mapping the IP contained the string `dns` or `ns` in an obvious fashion in the host part of the domain name, it is very likely a DNS server with a fixed address, so we keep it (local name servers usually require static addresses because clients must locate them in order to perform name resolution to begin with).

2. For the IPs that we were able to reverse map to domain names, we also reverse mapped the IP just above and just below it (e.g., for 128.2.0.10, we reverse mapped 128.2.0.9 and 128.2.0.11). We define the *difference* between two host names as the ratio of the number of different characters and the total number of characters, treating consecutive numeric characters (i.e. numbers) as single characters. If the difference between the host portion of the IP's domain name and the domain name just above and just below it was greater than 25%, then we keep it. Dynamic IPs are almost always from a block of addresses to which administrators assign very similar names (usually only alternating some digit for each address in the block). For example, when examining one IP in each of the 233,413 distinct class C address blocks that were in the spam RBL's list of dynamic IP pools and that reversed-mapped to a name, at least 98% were detected as dynamic using this heuristic (some of the remaining 2% appeared to be static IPs that just happened to fall within a class C network partially assigned to a dynamic IP pool; e.g., some had recognizable names like `mail` or `ns`).

3. Finally, we examined the actual sequence of probe responses from each LDNS server. If they were responsive to DNS and ping, then we know when one fails and the other succeeds. We hypothesize that if an LDNS server was using a dynamic IP, gave up the IP, and the IP was reused by another client in the pool, it is unlikely that the new client would happen to also be running a DNS server since client machines are rarely configured to serve DNS. Hence, for the servers that were responsive to both DNS and ping, we keep them if their DNS and ping response patterns were consistent during the entire period of the trace. In this case, even if the host was using a dynamic IP address, it is unlikely to have given it up during our measurement period.

We call this conservative estimate of "non-dynamic" LDNS servers `ldns-nondynamic`. This is the sample set that we use for the

[5] We count network failure within a NAC as an actual failure since service is unlikely to be available to anyone outside the NAC.

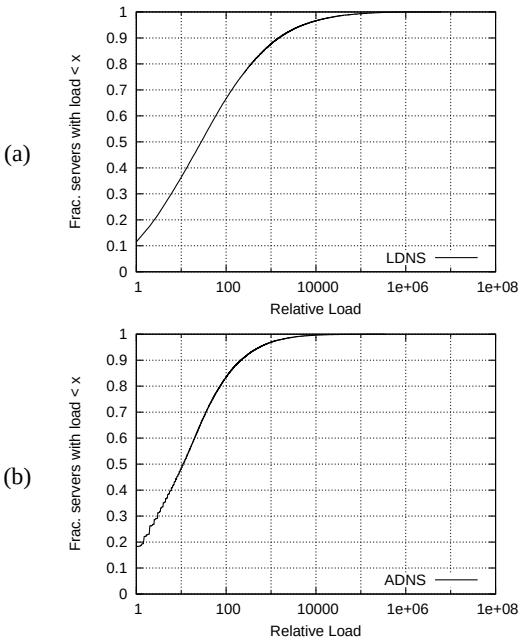


Figure 3: Relative Load Distribution: These figures show the CDF of the relative load of LDNS and ADNS servers (using `ldns-nondynamic` and `adns-web`, respectively).

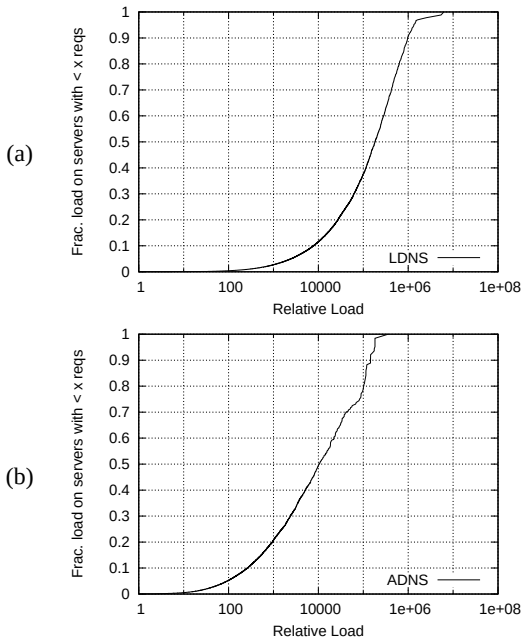


Figure 4: Aggregate Load Distribution: These figures show the fraction of total load that is accounted for by LDNS and ADNS servers with relative load less than x (using `ldns-nondynamic` and `adns-web`, respectively). For example, consider all LDNS servers that generated less than 10,000 requests each. Their aggregated load makes up about 10% of the total load seen by Akamai.

majority of our LDNS availability analysis, though we reexamine the dynamic IP issue in Section 4.4.

4. MEASUREMENT RESULTS

This section enumerates our measurement results and analyzes their significance. We first examine the relative load distribution on different nameservers, then perform an in-depth analysis of name server availability, failure and recovery times, and finally enumerate several common "deployment styles" that we observed in different types of organizations.

4.1 Relative Load Distribution

Figure 3 (a) and (b) show the cumulative distribution of relative load among LDNS and ADNS servers, respectively. Not surprisingly, the distribution is very skewed, with the majority of nameservers lightly loaded (i.e., generating or receiving less than 100 requests to Akamai or from the Web caches in a week's period), but a few that are much more heavily loaded (i.e., generating or receiving over 1 million requests). This is indicative of the diversity in the size of organizations behind individual LDNS servers (e.g., small businesses to large residential ISPs). It also suggests that the vast majority of nameservers have few users.

However, as Figure 4 (a) and (b) demonstrate, the "small" nameservers make up a very small fraction of the total load in the system. In the LDNS case, over 95% of servers made fewer than 10,000 requests to Akamai each, and their aggregate request count was only about 10% of the total number of requests generated by all the servers in the system. In the ADNS case, over 80% of servers received fewer than 100 requests from the web cache, and those requests constituted fewer than 5% of all requests sent. Hence, although most name servers are lightly loaded, most *users* are likely behind the smaller number of highly loaded nameservers. The distribution for LDNS servers does not quite obey a power law; highly ranked servers have relative loads within the same order of magnitude, possibly due to capacity limitations of the server machines, or due to more effective caching of low TTL records with extremely high request rates.

4.2 Availability

This section describe and analyze our measurements from the active probing experiment. Recall that during this experiment, we actively probed our set of LDNS and ADNS servers for a two-week period and recorded their responsiveness throughout the period. We begin by showing the overall availability numbers for LDNS and ADNS servers. Second, we discuss the correlation between availability and relative load. Third, we briefly describe the impact of the time of day on availability, and examine the extent to which failures are locally correlated. Then, we present a rough estimate of the time to failure and time to recovery of DNS servers. Finally, we revisit LDNS servers using dynamic IPs and estimate the arrival rate of new IPs they use in the system.

4.2.1 Overall Availability

We define the *availability* of a DNS server to be the ratio of the number of probes that it responded to and the total number of probes sent to it. Because the intervals between our probes were exponentially distributed, the PASTA principal dictates that we should see approximately the average fraction of time that a given server was available and unavailable.

Figure 5 summarizes the overall availability of LDNS and ADNS servers and plots the cumulative distribution of servers with a given availability. As is visible, the vast majority of both ADNS and LDNS servers are available for almost the entire duration of our experiment. In fact, 62% of LDNS servers and 64% of ADNS servers

Statistic	LDNS	ADNS
Mean Availability	0.9785	0.9864
Standard Deviation	0.0939	0.0701
Median	1.0000	1.0000
5th Percentile	0.9194	0.9532

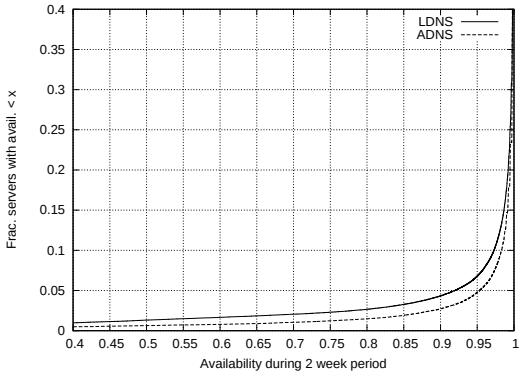


Figure 5: DNS Server Availability: The figure shows the distribution of availability of LDNS and ADNS servers (using `ldns-avail` and `adns-avail`). The table above it summarizes the availability statistics.

had 0 probe failures. Only 5% of LDNS servers (ADNS servers) were available for less than 92% (95%) of the time. As noted earlier, a few of these may be attributed to network failures, so the actual availability may be even higher. This should not be very surprising since these DNS servers are actively serving DNS requests and downtime may mean lack of service for an LDNS server's organization or an ADNS server's customers. However, we note that a non-trivial number of individual servers only have "one 9" (or less) of availability, a far cry from the "five 9s" usually desired for critical services.

We observe that ADNS servers have a slightly higher availability than LDNS servers in general. Keeping in mind that we are ignoring LDNS servers that may be behind dynamic IPs, the difference is probably higher in reality. In fact, when examining only ADNS servers in `adns-web` (that is, the ADNS servers authoritative for at least one website accessed by the web cache), the average availability is even higher, as the next section shows. This might indicate that content providers may be better equipped to keep DNS servers running (or that they have a greater incentive) than network administrators managing servers that only serve the internal population of an organization. Nonetheless, both ADNS and LDNS servers are supposed to be replicated [25], so a single DNS server failure does not necessarily imply that the service itself is unavailable.

4.2.2 Load vs. Availability

Next, we investigate whether the nameservers that are used more (i.e., are serving a larger population of clients) are more likely to have high availability than those that are used less. Recall that we defined the *relative load* on an LDNS server to be the number of A-requests that it sent to Akamai during a one week period. We defined the relative load on an ADNS server as the number of HTTP requests sent to websites it served that we observed in a one-week log trace of several web caches. We take load to be an approximation of server usage.

For our analysis, we only use LDNS servers that appear in both `ldns-nondynamic` and `ldns-load`, and ADNS servers that appear in both `adns-all` and `adns-load`, since we have both availability and relative load estimates for only these servers. Although the availability distribution of this subset of LDNS servers was not significantly different than that of all LDNS servers, the average availability of the ADNS servers examined in this analysis was higher than when examining the availability distribution for all ADNS servers in `adns-all`, as presented in the previous section (0.994 vs. 0.986). This observation is in line with our general conclusions, since presumably the DNS servers that did not appear in the web cache logs were rarely used.

Relative Load	Mean LDNS Avail.	Mean ADNS Avail.
0-100	0.978050	0.993850
100-1000	0.978989	0.996262
1000-10000	0.986182	0.996966
10000-100000	0.992636	0.998188
100000-1000000	0.995020	0.998639
$\geq$1000000	0.998795	NA
Correlation	LDNS	ADNS
corr($load, avail$)	0.017224	0.007867
corr($\log load, avail$)	0.041212	0.043248

Figure 7: Relative Load vs. Availability Summary: The top half of this table shows the average availability of LDNS and ADNS servers with given ranges of relative loads. The bottom half shows the correlation coefficient for relative load and availability and that for the log of the load and availability. Note that the load on LDNS and ADNS servers are estimates of different characteristics.

Figure 6 (a) and (b) show scatter plots comparing the relative load and availability of LDNS and ADNS servers, respectively. Clearly, there is a positive correlation between load and availability, especially in the region between 80% and 100% availability (we omit the 0% to 80% region in the ADNS case because there are very few points in that region). Figure 7 summarizes the results with the average availability of servers falling within a load range. Although the positive trend is readily apparent, the correlation is actually very minor; for example, an LDNS server that sent over 1 million requests to Akamai is only 1.02 times more likely to be available than an LDNS server that sent under 100.

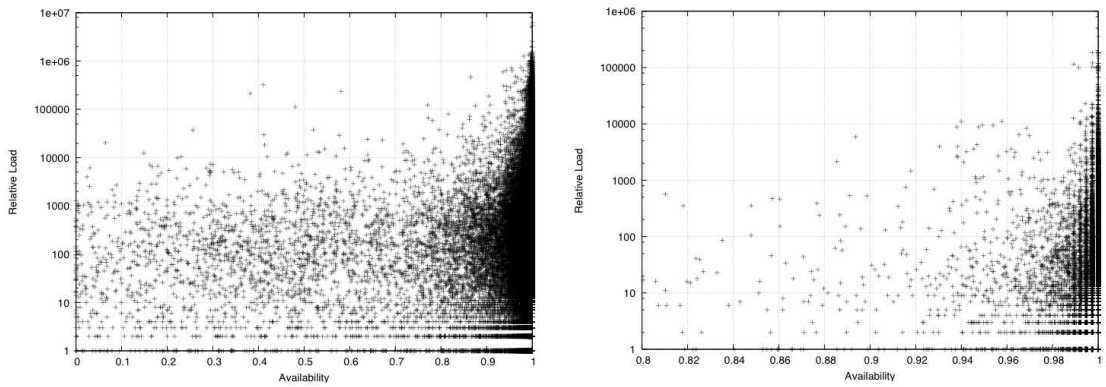
The bottom half of Figure 7 shows the correlation coefficient computed for relative load and availability over LDNS and ADNS servers. Here we see that the correlation is indeed very slight (0.017 for LDNS, and 0.008 for ADNS).[6] The correlation is more significant when we compute the coefficient using the logarithm of the relative load, indicating that the relationship between load and availability may be better described as log-linear than linear.

4.2.3 Time-of-Day Effects

Now we explore whether the time of day is related to when failures occur. We attempt to discern the *degree* of time-of-day impact on each server by comparing the availability during the hour in which it is most likely to be available to that during the hour in which it is least likely to be available. The ratio of the former and the latter gives us an indication about how much more likely a server will be available during an hour of the day. In this analysis, we only consider the fraction of servers with availability less than 1.

Figure 8(a) shows the cumulative distribution of servers with this statistic. About 70% of the servers are not more than 1.2 times as likely to be available in one hour than any other. Hence, for the majority of servers, time of day is unlikely to be related to when failures are observed. However, 2.1% (0.7%) of the LDNS (ADNS)

[6]Despite the small correlation coefficient, Pearson's product-moment correlation test indicates that the correlation is still statistically significant, given the number of sample points we have.



(a) LDNS Servers (b) ADNS Servers

Figure 6: Relative Load vs. Availability: These scatter plots show the relationship between the relative load on LDNS and ADNS servers and their availability (using the samples in the intersection of `ldns-all` and `ldns-avail`, and the intersection of `adns-web` and `adns-avail`, respectively). Note that the ranges of the x-axes on these graphs are different and the load on LDNS and ADNS servers are estimates of different values.

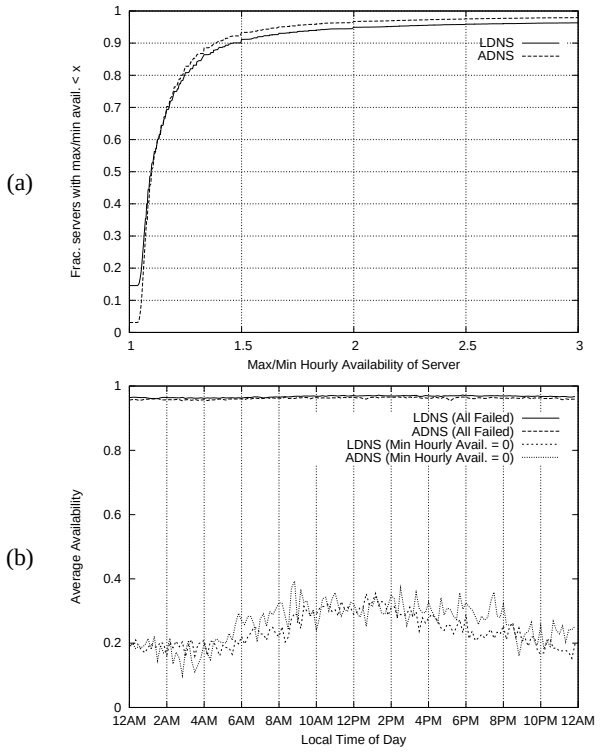


Figure 8: Time-of-Day Effects: (a) Shows a distribution of the degree to which time of day impacts availability. The *degree of impact* is defined as the ratio of the maximum average availability during a particular hour to the minimum, and indicates how much more likely a server will be available during one hour over another. (b) Shows the average availability during each 10 minute bucket in a day (adjusted to the server's local time) for all servers (the top two "All Failed" lines) and for servers with an infinite degree of impact (i.e., minimum availability is 0 for some hour). (These figures only use samples with < 1 availability in `ldns-avail` and `adns-avail`.)

servers had at least 1 hour with 0% availability (excluded from Figure 8(a)).

Figure 8(b) plots the average availability of servers during each 10 minute bucket in a day (adjusted to the server's local time[7]). The top two lines show the variation of the averages for all nameservers (that had availability < 1), and the bottom two show the variation of the averages for nameservers that had at least one hour with 0% availability. There is almost no variation in the average availability when looking at all nameservers. However, when looking only at those that are very likely to be unavailable during some hour of the day, we observe a clear diurnal pattern. In addition, these servers have much lower overall availability. We believe it is likely that these LDNS "servers" may be on end-user machines (though this would not explain the small number of ADNS servers that behave this way). This observation also lends evidence to our hypothesis that many LDNS "servers" are behind DSL and dialup addresses.

4.2.4 Locally Correlated Failures

Next we examine the degree to which failure periods are correlated within particular domains. Here we define a domain to be a NAC [19] to capture events correlated with network location. Hence, correlated failures suggest common network outages that prevent reachability or other related events within a NAC (power outage, network-wide maintenance, etc.).

We estimated the fraction of failure probes that are correlated within a NAC as follows: first we examined all *locally-related* probe pairs — closely spaced probe pairs (within 2 minutes) that contained at least 1 failure, and were sent to different servers within the same NAC. Of the NACs formed by the servers in `ldns-avail` (`adns-avail`), 20% (11%) had at least one such pair. We call the pair *correlated* if both probes were failures. The ratio of correlated pairs and total locally-related pairs gives an estimate of the fraction of failures within the NAC that are correlated. The cumulative distribution of this fraction (over all NACs) is shown in Figure 9.

We observe that about 40% (50%) of LDNS (ADNS) NACs have no failures correlated in this manner. However, on average 11.5% (12.2%) of failures were correlated in a given NAC. This roughly corresponds to the fraction of probes to LDNS servers that we es-

[7]We used Edgescape [3], a commercial tool, to classify the location of IPs and their timezone. We ignored IPs that it could not classify in our analysis.

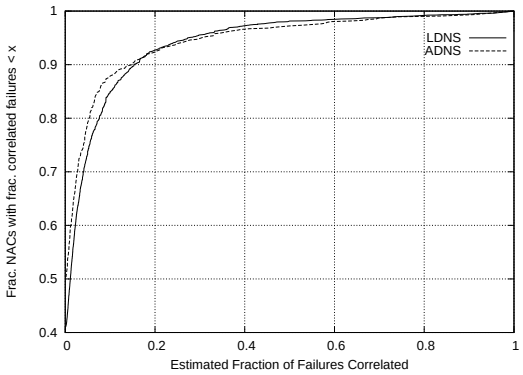


Figure 9: Correlated Failures within NACs: This figure shows the estimated fraction of failure probes to different servers within each NAC that were "correlated;" i.e., they occurred within 2 minutes of each other (using `ldns-avail` and `adns-avail`; see text).

timated may have been affected by network failures in Section 3.3 (about 15%). In addition, there are only a few NACs (less than 10%) that have a large fraction ($> 20\%$) of failures that are correlated. Thus, most unavailability to DNS servers is unlikely to be correlated within a NAC.

4.3 Time to Failure and Recovery

So far we have explored the availability of DNS servers — the probability that a given DNS server is available during a point in time. In this section, we give a very rough characterization of how long we expect a DNS server to be available before it fails and how long we expect it to take until it recovers. This information is difficult to extract from our measurements directly because we might miss failure or availability periods due to the granularity of our probes. In particular, we note that a large fraction of the failures we observed only occurred for a single probe. Further complicating the analysis is the fact that we cannot distinguish network failures (which may still be present in our data despite our filtering heuristics) and actual DNS server failures.

Thus, in this analysis, we present two sets of results: one that ignores any failure that we cannot determine to be longer than 20 minutes ("short failures"), and one that includes all failures. Feamster *et al.* [12] found that more than 90% of all network failures lasted less than 15 minutes, so server failures can be accurately estimated from probe failures in our former set of results because network failures are almost always of short duration. Moreover, during our high-frequency probing of a small subset of servers (one probe per 5 minutes; see Section 3.2), we found that more than 60% of all observed failures periods encompassing more than a single probe lasted longer than 20 minutes (for both ADNS and LDNS servers); hence, most failures we observed that were longer than 5 minutes were also longer than 20 minutes. However, the results ignoring short failures will tend to overestimate the time to failure and overestimate the time until recovery. The results including all failures present a more pessimistic estimate of the time to failure.

We estimated the time to failure as the length of time from when we first probed a DNS server until we observed the first probe that fails (possibly ignoring short failures as described above). We estimated the time to recovery as the length of time from the start of the first probe failure until the last consecutive probe failure. Since we optimistically assume that the failure does not begin until the probe failure, our results may underestimate failure lengths by an amount within an order of magnitude of the mean probing interval (1 hour), on average. One issue with this analysis is that we may have missed failure or availability periods if they are likely to last shorter than our average probing interval.

To investigate the degree to which short failures might affect our results, we examine "closely spaced" probe triples that we made to DNS servers. Suppose S represents a probe success and F represents a probe failure in a sequence of consecutive probes. Then, when looking at probe triples made within a 2 minute period to LDNS servers, 0 triples had a pattern of F S F, and 14 of 40,322 (0.03%) triples had a pattern of S F S; hence, it is unlikely that failure or availability periods last less than 2 minutes. When looking at 30 minute intervals, 1,401 and 6,634 of 7,041,334 (0.02% and 0.09%) triple samples had a pattern of F S F or S F S, respectively; however, the short availability periods were primarily isolated to 934 servers (i.e., only these servers had a number of short availability periods much greater than the mean). Closely spaced samples at ADNS servers had similar characteristics.

Examining longer periods yields similar fractions that are short failure periods, but not larger fractions that are short availability periods. Hence, since the probability of short availability periods is low, we are at worst likely to overestimate the length of very long consecutive availability periods (due to short failures that our probes missed). Nonetheless, we caution that our estimates should be taken as very coarse approximations.

Statistic	Ignoring Short Failures		With Short Failures	
	LDNS	ADNS	LDNS	ADNS
$\geq$1 Failure	12.6%	10.8%	37.8%	35.7%
For fraction with at least 1 failure (ignoring the 5% with the highest values)				
Mean Time to Failure	125.9 h	143.1 h	132.4 h	148.7 h
Standard Deviation	99.1 h	100.0 h	98.3 h	99.1 h
Median	106.6 h	134.0 h	117.3 h	138.0 h
Mean Time to Recovery	7.2 h	6.3 h	NA	NA
Standard Deviation	9.5 h	8.7 h		
Median	3.3 h	2.6 h		

Figure 10: Failure and Recovery Summary: This table shows failure and recovery statistics for the fraction of LDNS and ADNS servers for which we observed at least one failure, with and without heeding failures that we could not determine were longer than 20 minutes (statistics are taken using `ldns-avail` and `adns-avail`, ignoring the 5% with the highest values).

Figure 10 presents summary statistics about the time to failure and recovery for when we ignore short failures and when we don't. Clearly, the number of servers where we observe at least one failure is much larger when we do not ignore the short failures (37.8% vs. 12.6% for LDNS servers; 35.7% vs. 10.8% for ADNS servers). However, we note that including all failures drives up the mean time to failure among those that did fail by several hours in both cases (hence, servers that we observed short failures at tended to fail later than those that we observed longer failures at), and well more than the majority never fail at all during the two-week period in both cases. We omit statistics for recovery times when including all failures, since many of these failures lasted for only a single probe, and we cannot conclude much about how long the failure lasted.

Figure 11 plots the cumulative distribution of time to the first failure for those servers that were unavailable at least once (ignoring the 5% with the highest values to limit outliers). Whether or not we take into account "short" failures, the majority of servers are available uninterrupted for longer than two weeks. Our measurement period was not long enough to make longer term conclusions, except that the time to failure for DNS servers is (at least)

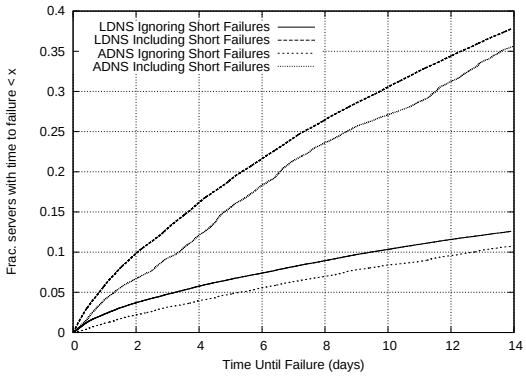


Figure 11: Time To Failure: This figure shows the distribution of times until the first failure for the fraction of LDNS and ADNS servers that failed at least once during the 2 week observation period (ignoring the 5% with the highest values). The plot shows the time to failure with and without heeding failures that we could not determine were longer than 20 minutes. (Our measurement period lasted 2 weeks so the figure ends at 14 days.)

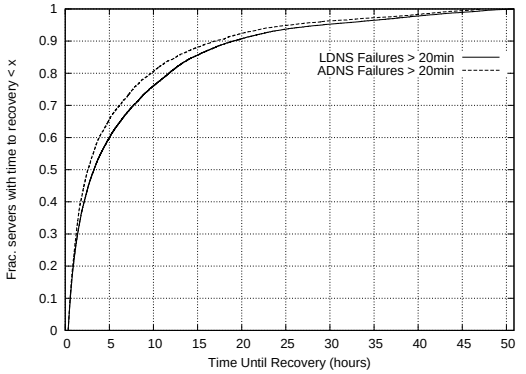


Figure 12: Time to Recovery: This figure shows the distribution of times until the first recovery for the fraction of LDNS and ADNS servers that failed at least once and then recovered during our 2 week period (ignoring the 5% with the highest values).

on the order of days or weeks. ADNS servers are likely to live a little longer than LDNS servers, which is expected given that they are more highly available.

Figure 12 plots the cumulative distribution of the time between the first server failure and when the server recovers, for all servers that failed and recovered during our two-week period (again, ignoring the 5% with the highest values). We observe that the majority of failures last less than 5 hours. Considering that we are ignoring failures that we could not determine lasted more than 20 minutes, that fraction is likely to be larger. Less than 7% of the observed failures last longer than 1 day. Hence, recovery times are likely to be on the order of hours, and very unlikely to last more than 2 days. Again, ADNS servers have slightly shorter recovery times than LDNS servers.

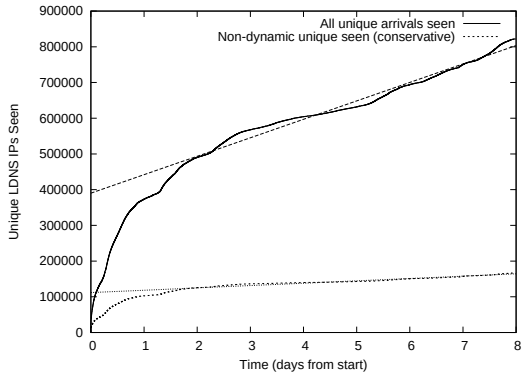


Figure 13: LDNS Arrival Rate: This figure shows the number of unique LDNS IPs that accessed a portion of Akamai (upper curve), and the estimated number of unique non-dynamic IPs that accessed Akamai (lower curve) during a one-week period on a subset of Akamai nodes (using `ldns-all`). The overlaid dashed lines approximates the arrival rate of new IPs after 1.5 days.

4.4 Dynamic IP LDNS Arrival Rate

We now revisit the LDNS servers that we believe may be behind dynamic IP addresses. In particular, based on reverse DNS lookups and by cross-checking IPs with a spam RBL database, we noticed that a fair number of LDNS servers appear to be on DSL or dialup links. Due to the transient nature of machines behind dynamic IP addresses, we cannot derive much about them from our active probes. Instead we attempt to infer their attributes by characterizing their time of appearance at Akamai.

Figure 13 shows the arrival rate of new unique IP addresses to the subset of Akamai regions for which we collected logs. After 1.5 or 2 days, it appears that the rate of new arrivals becomes roughly linear. Measurements of LDNS servers accessing the root nameservers have also exhibited this pattern [8]. It is highly unlikely that a large number of new LDNS servers are continually being added to the DNS system, so we suspect the majority of these arrivals are from DNS servers that change their IP addresses. We also plot the arrival rate of LDNS servers that we conservatively classify as non-dynamic (using the first two heuristics described in Section 3.3). We used linear regression to fit lines to the new arrivals after 1.5 days, which is shown as the dashed and dotted lines in the figure. The the total arrival rate line has a slope of 51,666, while the slope of non-dynamic arrival rate line has a slope of 6,572 (a factor of 8 smaller!), suggesting that the arrival rate of new dynamic IPs to this subset of Akamai regions is roughly 31 per minute. Since this subset is composed of roughly one eighth of Akamai's regions, the global arrival rate of LDNS servers on dynamic IPs may be much higher.

This arrival rate is an overestimate, since our non-dynamic classification heuristics are conservative (observe that if most servers that appear during the first 1.5 days are using non-dynamic IPs, then our estimate may be off by a factor of 4). However, arrival rates differ by a factor of 8, so we can still conclude that the the actual arrival rate of new dynamic IPs is still very high (though eventually the dynamic IP pools will be exhausted). Unless explicitly guarded against, future federated services may have to manage these "servers" in addition to the aliasing effects of dynamic IPs. The LDNS deployment model allows anyone to establish their own local nameserver, and it appears that "anyone" will, whether intentionally or not.

4.5 Deployment Styles

We conclude our measurement results by providing a preliminary examination of how particular organizations deploy and configure their various DNS servers to respond to local queries – i.e., how many DNS servers do they deploy and how the load distributed across these servers. We refer to this as the LDNS *deployment style*. For example, some organizations may just deploy a single local DNS server to answer all local queries. Other organizations could deploy a collection of local DNS servers each serving a fraction of the request volume. For this analysis, we used a trace of DNS requests made to 80% of active Akamai servers during one day in December 2003, similar to that used to derive `ldns-load` (the larger coverage of Akamai sites in this trace allows us to obtain a more complete view of organizations that access Akamai).

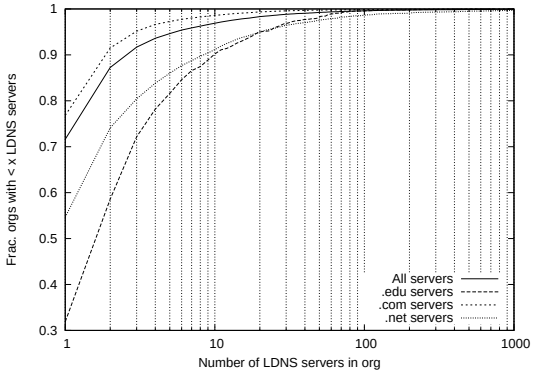


Figure 14: Number of LDNS servers: A CDF of the number of LDNS servers observed within each organization. The x-axis is in log-scale. Note that the all servers, .net, and .com lines extend beyond the right-edge of the graph; in particular, the .net line does not reach 1 until about 11,000.

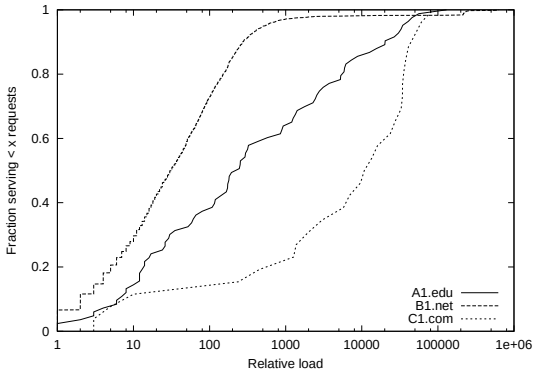


Figure 15: Deployment examples: A CDF of the relative load on LDNS servers (i.e., number of requests made by the server) for three different organizations. The x-axis is in log-scale. The number of LDNS servers in A1.edu, B1.net and C1.com are 84, 1710 and 27, respectively.

Figure 14 shows a CDF of the number of LDNS servers we observed in all organizations. Although most organizations appear to have only a single LDNS server deployed, .edu organizations are clearly more likely to have multiple servers. However, .net organizations, which are typically ISPs, are the most likely to have a very large number of LDNS servers (more than 50). Note that we can not distinguish multiple LDNS servers behind a firewall or NAT device; this limitation may contribute to the smaller number of LDNS servers observed in .com organizations, since presumably they are more likely to employ such devices.

Figure 15 shows the relative load distributions across LDNS servers (including dynamic ones) in three different organizations, a .com, a .net, and a .edu. We use the domain name of an LDNS to identify the organization that it "belongs to". For a .edu (universities) or .com (enterprises) site, this approach will only identify LDNS servers that are deployed and maintained by the site. However, this does not hold true for .net organizations, which are typically ISPs and backbone carriers. For example, an LDNS server `dynip-1-2-3-4.B1.net` will be classified as "belonging" to `B1.net`, while it probably belongs to a small customer of `B1.net`. In the case of .net organizations, therefore, our approach identifies nameservers whose traffic is carried by the ISP and which are still dependent on the ISP for connectivity and persistent availability.

As Figure 15 shows, a large number of LDNS servers in `B1.net` generate very few requests (80% of servers generate less than 100 requests to Akamai during the course of the day). Also, a very small fraction of LDNSs generate a large number of requests (2% of LDNSs each generate 100,000 requests to Akamai during the day). At the other extreme, a much smaller fraction of LDNS servers in `C1.com` are lightly loaded (only about 15% of LDNS servers generate less than 100 requests to Akamai each). Also, there exist a few servers that generate intermediate and very high request volumes. Finally, the curve for servers in `A1.edu` is in between these two extremes with a much wider variety of relative loads across the servers.

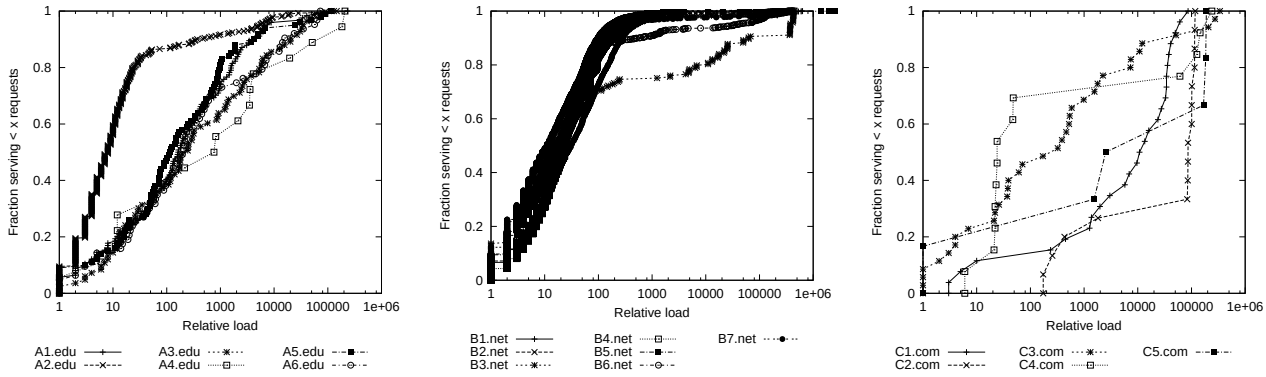
In Figure 16, we show other examples of relative LDNS loads for well known .edu (Figure 16(a)), .net (Figure 16(b)) and .com (Figure 16(c)) sites. Notice that, with the exception of `A2.edu`, all .edu sites in Figure 16(a) roughly follow the same trend as `A1.edu` in Figure 15. `A2.edu` likely shows different behavior because it shares DNS servers with a local ISP. The trend among the different .net sites (Figure 16(b)) is again similar to that of `B1.net` in Figure 15. However we do not observe as clear a common pattern among the different .com sites (Figure 16(c)). For example, while `C2.com` and `C1.com` have similar characteristics, `C3.com` is more similar to the style we observe for .edu sites.

In general, the load distribution among an organization's deployed LDNS servers seems to belong to one of three broad varieties as we show in Figure 15. In the future, we plan to further investigate the underlying trade-offs that lead to these classes of deployment.

4.6 Summary of Measurements

What conclusions can we draw from our observations about DNS server characteristics? First, we conclude that the majority of users are likely to be using only a small number of the LDNS servers deployed. The majority of DNS requests sent over the wide area are for a small number of ADNS servers. These results imply that the distribution of user populations behind particular servers in DNS is highly skewed.

Second, we observe from our results that the majority of both LDNS and ADNS servers are highly available: most were always available during our two week observation. For those that had unavailability periods, time to failure typically lasted days, weeks, or longer, and recovery times typically lasted less than a few hours. A non-trivial fraction had "one 9" or less of availability, but with replication at independent points in the network, we believe DNS service is almost always available. Moreover, servers that are used



(a) .edu servers (b) .net servers (c) .com servers

Figure 16: DNS server deployment styles: The figures show the CDF of the relative load on LDNS servers (i.e., number of requests made by the server) for three different types of organizations: .edu sites (universities, (a)), .net sites (Internet service providers and backbone carriers, (b)) and .com sites (commercial enterprises, (c)).

more frequently are more likely to be available for a larger fraction of time. Although the correlation between frequency of use and availability is probably not large enough to make predictive estimates, it does bode well considering the significant skew in usage across nameservers.

Third, we find that there is significant diversity in deployment styles within different organizations. Some organizations, such as ISPs, are comprised of both very highly used and very lightly used nameservers, while others, such as commercial enterprises, only have several medium and highly used servers. The deployment style may reflect the departmentalization within organizational boundaries or load balancing among multiple servers.

Finally, we make the observation that LDNS servers have much more diversity in their attributes than ADNS servers. For example, we found that about 2% of the LDNS servers that had less than perfect availability exhibited diurnal patterns of availability and that a large number of "new" LDNS servers appeared to be arriving over time, as it appears that many are using dynamic IP addresses, many of which are behind DSL, cable, or dial-up links. This diversity probably contributes to the higher availability that we observe in ADNS servers and suggests that unless guarded against, infrastructures deployed in a manner similar to LDNS servers will face more heterogeneity than those deployed like ADNS servers.

5. APPLICATION TO DHTS

As an example of applying DNS measurements to the design of a federated distributed system, we model the effect of dedicated infrastructure with similar availability characteristics as DNS on a Distributed Hash Table (DHT).[8] A DHT is a fundamental building block in several proposed Internet-scale infrastructures [18, 20, 28, 33, 35]. In such DHT-based systems, availability governs the degree of *churn*, which impacts stability and performance [21, 29]. Figure 17 shows the setup of our simulation, which is similar to that of the study conducted by Li *et al.* [21]. Infrastructure servers could be deployed in a federated manner similar to LDNS — for example, an infrastructure node could be used as a "well-known node" that directs initial queries, thus serving a role similar to an LDNS server.

[8]Note that we do not propose that DNS should be run on a DHT (though others have proposed so), only that other services implemented using a DHT might deploy infrastructure in a similar fashion to that of DNS.

Parameter	Setting and Description
Environment	1024 nodes simulated in p2psim [26].
Network topology	Pairwise latencies of 1024 DNS servers collected using the King method [14] by Li *et al.* [21].
DHT algorithm	Chord [34] with proximity neighbor selection.
Lookup rate	1 per node using exponentially distributed arrival intervals with mean 2 minutes.
Time to Failure and Recovery	
Client nodes	Modeled after clients seen in an operational peer-to-peer file-sharing network [13]: Exponentially distributed time to failure with mean 2.4 minutes. Time to recovery is also exponentially distributed with mean 2.4 minutes.
Server nodes	Modeled after LDNS servers measured in our study: 37.8% use the empirical LDNS time to failure distribution (see LDNS with short failures data in Figure 10); the rest of the servers were never seen to fail, so we pessimistically model them as failing with an exponentially distributed mean time to failure of two weeks. All use the empirical LDNS time to recovery distribution.

Figure 17: DHT simulation setup: Parameters used in our DHT simulations.

Using parameters derived from our DNS measurements, we show how dedicated infrastructure nodes can significantly reduce the overhead bandwidth in a DHT. Overhead bandwidth comes in two forms: lookup traffic and maintenance traffic. Lookup traffic scales proportionally to the lookup rate and to the log of the size of the network in Chord. Maintenance traffic depends on the maintenance interval, the inverse of how often a node checks that all of its neighbors are still alive and repairs broken overlay links — longer intervals incur less maintenance traffic. If nodes leave frequently, the maintenance interval must be shorter in order to prune stale routing data. Likewise, if some nodes are part of a dedicated infrastructure, the interval can be longer. Efficiency mandates a maintenance interval inversely proportional to the lookup rate or longer. But reliability requires the maintenance interval to be proportional to the average lifetime or shorter. Hence, when the lookup rate and lifetime are inversely proportional, extending the lifetime will substantially decrease maintenance traffic.

Figure 18 shows our simulation results, varying the fraction of dedicated infrastructure nodes (*servers*) and end-user nodes (*clients*) in the DHT. With a portion of nodes acting as dedicated infrastructure, we can achieve similar reliability while decreasing bandwidth (or maintain bandwidth and improve reliability). For example, if a quarter of the nodes are servers rather than clients, the network requires roughly half the bandwidth to achieve similar reliabilities.

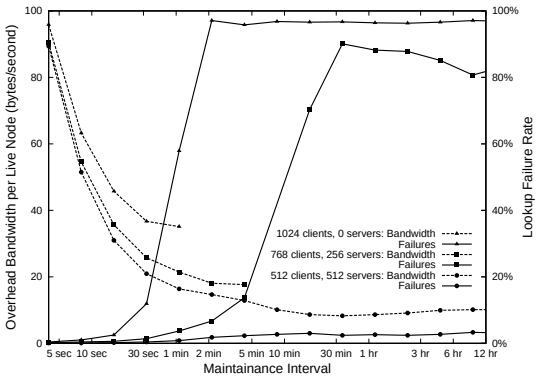


Figure 18: Simulation results: This figure shows how varying the maintenance interval impacts the amount of overhead traffic incurred by each node in the DHT (dashed lines), and the rate of failed lookups (solid lines). We vary the fraction of dedicated infrastructure nodes (*servers*) and end-user nodes (*clients*) in the DHT. Results from a DHT composed completely of server nodes (1024 servers, 0 clients) are omitted for clarity, but would have less than 0.2% lookups fail even with a maintenance interval of 12 hours and would follow the downward trend of overhead traffic.

Our simulation results show that having infrastructure with failure properties no better than that of LDNS servers allows DHT maintenance to be performed much less frequently while still achieving a high rate of successful lookups. Other observations from our measurement study may also have implications for DHTs. For example, we saw that the number of requests generated by LDNS servers was highly skewed; hence in a federated DHT, there may a few nodes that generate the majority of the lookups. In such a scenario, measures might need to be taken in order to more fairly balance the load required to route these lookups. In addition, certain deployment styles within organizations may be amenable to more hierarchical overlay construction. We leave this for future work.

6. RELATED WORK

In this section, we provide brief surveys of related work on DNS characterization, availability measurements, and the impact of churn on DHT performance.

6.1 DNS

Danzig *et al.* [11] presented the first comprehensive study of DNS traffic in the wide area. Using measurements at a root name server, they found a large number of misbehaving local DNS resolvers and analyzed the characteristics of DNS traffic. Brownlee *et al.* [6, 7] discuss more recent measurements taken at the root name servers and continue to find a large number of illegitimate queries in DNS traffic. Liston *et al.* [22] studies DNS traffic from the client vantage point and analyzes their diversity across sites. Jung *et al.* [17] also examined DNS traces from two local DNS servers and analyzed the impact of caching of A-records in DNS as well as the setting of TTLs on client performance. Our study, in contrast, looks primarily at the characteristics of the DNS infrastructure instead of the particular traffic characteristics. Ramasubramanian and Sirer [28] also examined characteristics of the authoritative DNS hierarchy, such as the prevalence of bottlenecks in name resolution and the number of nameservers containing known security vulnerabilities.

Both Shaikh *et al.* [32] and Mao *et al.* [24] analyzed the proximity of clients to their local DNS servers. They found that a fair number of clients were not close to their local DNS server and their performance could be improved by using a more proximal server, such as one in their network aware cluster. Cranor *et al.* [10] looked at the distribution of ADNS and LDNS servers found by looking at DNS traffic and grouped them using network aware clustering. We performed a similar analysis by clustering based on domain names, which are likely to reflect administrative domains.

6.2 Availability

Several recent studies [30, 5, 31] have analyzed the availability of participants of peer-to-peer file-sharing systems. Long *et al.* [23] studied the availability of hosts on the Internet, and their study is perhaps the most similar to ours; however, we focus on the availability of dedicated infrastructure and our measurements are much more recent (their study was conducted in 1995).

6.3 DHTs and Churn

Li *et al.* [21] and Rhea *et al.* [29] have examined the impact of churn on DHTs and devise mechanisms for managing extremely low mean time to failures. Our evaluation suggests that such mechanisms are unnecessary in an infrastructure based system because the infrastructure allows for very low maintenance traffic exchange rates.

7. SUMMARY

In this paper, we presented measurements of a large number of local and authoritative DNS servers and analyzed their load, availability, and deployment characteristics.

Our key findings are that a large fraction of all end-users use a small number of local DNS servers. We found that a significant fraction of local DNS servers are highly available, without a single observed failure, with authoritative servers generally having higher availability. We found evidence that there is a slight positive correlation between usage and availability of DNS servers. We also observed a large degree of diversity in local DNS server deployment and usage: many servers originated from dynamic IP addresses pools. Also, some servers exhibited diurnal availability patterns. Finally, we observed that local DNS server deployments within organizations also tend to be diverse, ranging from a very few highly used servers to a very large number of lightly loaded ones.

Our observations shed new light on characteristics of DNS infrastructure. They are are also important to the study of future infrastructure services deployed in a federated manner. For example, we simulated a Distributed Hash Table using availability characteristics similar to DNS and showed how much infrastructure support improves reliability and decreases overhead.

Acknowledgments

We would like to thank Hui Zhang for initial feedback and our anonymous reviewers for their valuable feedback and suggestions. We would like to thank Steve Hill, Eric Olson, and Arthur Berger at Akamai for helping us with our infrastructure and Jeremy Stribling for helping us with p2psim. James Hendricks is supported in part by an NDSEG Fellowship, which is sponsored by the Department of Defense.

8. REFERENCES

[1] Not just another bogus list. http://dnsbl.njabl.org.

[2] University of Oregon, RouteViews Project. `http://www.routeviews.org`.

[3] Akamai Technologies. Edgescape. `http://www.akamai.com/en/html/services/edgescape.html`, 2004.

[4] G. Ballintijin, M. van Steen, and A. S. Tanenbaum. Scalable user-friendly resource names. *IEEE Internet Computing*, 5(5):20–27, 2001.

[5] R. Bhagwan, S. Savage, and G. M. Voelker. Understanding availability. In *2nd International Workshop on Peer-to-peer systems*, December 2002.

[6] N. Brownlee, k claffy, and E. Nemeth. DNS measurements at a root server. In *Globecom*, 2001.

[7] N. Brownlee, k claffy, and E. Nemeth. DNS Root/gTLD performance measurements. In *Proc. Passive and Active Measurement workshop (PAM)*, 2001.

[8] Clients of DNS root servers. http://www.caida.org/ kkeys/dns/, 2002.

[9] R. Cox, A. Muthitacharoen, and R. Morris. Serving DNS using Chord. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, Cambridge, MA, March 2002.

[10] C. D. Cranor, E. Gansner, B. Krishnamurthy, and O. Spatscheck. Characterizing large DNS traces using graphs. In *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement*, pages 55–67, 2001.

[11] P. Danzig, K. Obraczka, and A. Kumar. An analysis of wide-area name-server traffic. In *Proceedings of the SIGCOMM '92 Symposium on Communications Architectures and Protocols*, 1992.

[12] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of internet path faults on reactive routing. In *Proceedings of the ACM Sigmetrics 2003*, pages 126–137, 2003.

[13] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Modeling and analysis of a peer-to-peer file-sharing workload. In *Proceedings of the 19th Symposium on Operating System Principles*, October 2003.

[14] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating latency between arbitrary Internet end hosts. In *Proceedings of Internet Measurement Workshop 2002*, pages 5–18, 2002.

[15] IRCache. `http://www.ircache.net/`.

[16] ISC Internet Survey. `http://www.isc.org/ops/ds/new-survey.php`.

[17] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS performance and the effectiveness of caching. *IEEE/ACM Trans. Netw.*, 10(5):589–603, 2002.

[18] B. Karp, S. Ratnasamy, S. Rhea, and S. Shenker. Spurring adoption of DHTs with OpenHash, a public DHT service. In *Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, February 2004.

[19] B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. In *Proceedings of the SIGCOMM '00 Symposium on Communications Architectures and Protocols*, pages 97–110, 2000.

[20] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. Oceanstore: An architecture for global-scale persistent storage. In *Proceedings of the Ninth International ACM Conference on Architectural Support for Programming Languages and Operating Systems*, November 2000.

[21] J. Li, J. Stribling, T. M. Gil, R. Morris, and M. F. Kaashoek. Comparing the performance of distributed hash tables under churn. In *Proc. of the 3rd IPTPS*, February 2004.

[22] R. Liston, S. Srinivasan, and E. Zegura. Diversity in DNS performance measures. In *Proceedings of Internet Measurement Workshop 2002*, pages 19–31, 2002.

[23] D. D. E. Long, A. Muir, and R. A. Golding. A longitudinal survey of Internet host reliability. In *Proc. Symposium on Reliable Distributed Systems*, pages 2–9, 1995.

[24] Z. M. Mao, C. D. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A precise and efficient evaluation of the proximity between web clients and their local DNS servers. In *Proceedings of the USENIX 2002 Annual Technical Conference*, pages 229–242. USENIX Association, 2002.

[25] P. V. Mockapetris. Domain names - concepts and facilities. Request for Comments 1034, Internet Engineering Task Force, November 1987.

[26] P2Psim. `http://pdos.lcs.mit.edu/p2psim`.

[27] V. Paxson. End-to-end routing behaviour in the internet. In *Proceedings of the SIGCOMM '96 Symposium on Communications Architectures and Protocols*, September 1996.

[28] V. Ramasubramanian and E. Sirer. The design and implementation of a next generation name service for the Internet. In *Proceedings of the SIGCOMM '04 Symposium on Communications Architectures and Protocols*, August 2004.

[29] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling Churn in a DHT. In *Proceedings of the USENIX 2004 Annual Technical Conference*, June 2004.

[30] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking (MMCN'02)*, January 2002.

[31] S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. In *Proc. ACM SIGCOMM Internet Measurement Workshop (IMW)*, 2002.

[32] A. Shaikh, R. Tewari, and M. Agrawal. On the effectiveness of DNS-based server selection. In *Proceedings of the IEEE INFOCOM 2001*, Anchorage, Alaska, 2001.

[33] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *Proceedings of the SIGCOMM '02 Symposium on Communications Architectures and Protocols*, August 2002.

[34] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the SIGCOMM '01 Symposium on Communications Architectures and Protocols*, 2001.

[35] M. Walfish, H. Balakrishnan, and S. Shenker. Untangling the web from DNS. In *Proceedings of the First Usenix Symposium on Networked System Design and Implementation (NSDI'04)*, San Francisco, CA, March 2004.

[36] R. Wolff. Poisson arrivals see time averages. *Operations Research*, 30(2):223–231, 1982.