On the DNS Deployment of Modern Web Services

Shuai Hao*†, Haining Wang*, Angelos Stavrou‡, and Evgenia Smirni†
*University of Delaware, Newark, DE, USA
†College of William and Mary, Williamsburg, VA, USA
‡George Mason University, Fairfax, VA, USA
Email: {haos,hnw}@udel.edu, astavrou@gmu.edu, esmirni@cs.wm.edu

***Abstract*—Accessing Internet services relies on the Domain Name System (DNS) for translating human-readable names to routable network addresses. At the bottom level of the DNS hierarchy, the authoritative DNS (ADNS) servers maintain the actual mapping records and answer the DNS queries. Today, the increasing use of upstream ADNS services (i.e., third-party ADNS-hosting services) and Infrastructure-as-a-Service (IaaS) clouds facilitates the establishment of web services, and has been fostering the evolution of the deployment of ADNS servers. To shed light on this trend, in this paper we present a large-scale measurement to study the ADNS deployment patterns of modern web services and examine the characteristics of different deployment styles, such as performance, life-cycle of servers, and availability. Furthermore, we focus specifically on the DNS deployment for subdomains hosted in IaaS clouds.**

I. Introduction

As a hierarchical distributed database system, the Domain Name System (DNS) is one of the most important components of Internet infrastructure, providing the mapping between the domain names and network-level addresses to direct clients to specific Internet services. In DNS hierarchy, the Root and Top-Level-Domain nameservers are mainly used as the querying referrals, while the authoritative DNS (ADNS) servers, administered by the service providers, are responsible for storing the name-to-address records and returning answers to the clients.

Deploying authoritative nameservers requires extra hardware resources and additional maintenance support. Also, the critical roles of DNS service in web infrastructure make it an attractive target to attackers. Thus, web service providers are increasingly adopting the upstream authoritative DNS servers, including the top sites (e.g., Amazon and Twitter) that have the ability to maintain their own ADNS infrastructures. In addition, to save a large amount of investment for infrastructure, many of today's popular web services are directly built upon Infrastructure-as-a-Service (IaaS) clouds such as Amazon EC2 and Windows Azure. The traditional web service providers are also migrating extended services into clouds to use the "illusively-infinite" computing and storage resources. The IaaS infrastructure greatly facilitates the establishment of modern web services and also promotes the process of delegating the authoritative name resolution to third-party ADNS service providers. Besides traditional web-hosting providers such as Dyn [6] and Ultradns [14], the Content Delivery Networks (CDN) and cloud service providers also offer the ADNS services that integrate the name resolution into their CDNs or cloud infrastructures [1], [4].

Existing DNS measurements studied the characteristics of DNS activities and operations [16], [17], [21], [24], [26], the root or top-level-domain servers [20], [22], [29], [30], [36], or the DNS resolvers [15], [18], [35]. Some works involving the characteristics of ADNSes mainly focused on the comparison with local DNS (LDNS) servers, but none of them explored various ADNS deployments for web services. Complementary to these prior works, we present a large-scale measurement study in attempt to answer the following questions: (1) how do modern web services deploy their ADNS servers? (2) what are the characteristics of different ADNS deployment patterns? and (3) in particular, how do the cloud-hosting subdomains administer their ADNS servers?

We first collect the authoritative DNS server information for top-ranking websites on Alexa's list [2] and eliminate the redundant domain records. This constructs our dataset with about 2.3 million nameservers for about 0.94 million websites. We then develop a systematic method to explore ADNS server deployment patterns and perform the geo-distributed probing experiments. In particular, by directly issuing DNS queries to each ADNS server, we examine their deployment details and characteristics. Next, we focus on the DNS deployment of web services whose subdomains are hosted in cloud infrastructure. We extract the subdomain list from an existing dataset [5], reproduce the ADNS servers of subdomains for comparing with the original results, and examine their deployment. We summarize our major findings and contributions as follows:

- We use a simple heuristic method to determine the ADNS deployment patterns. In fact, it is fairly easy to recognize the pattern for an individual website from its NS records, but it is much more difficult when looking for millions of websites in such a large-scale study.
- We validate the use of ADNS proxy infrastructure by examining the transition delay and the TTL aging.
- We first quantify the usage and profile the characteristics of ADNS servers in terms of the deployment patterns.
- We find that most top-ranked websites deploy their own DNS servers but emerging popular social sites tend to use the upstream DNS-hosting services. We also observe few servers being used in private deployment.
- We find that the ADNS deployment patterns remain stable. The change of private servers is more frequent than that of upstream servers. The websites using upstream services change frequently their hosting domains but have

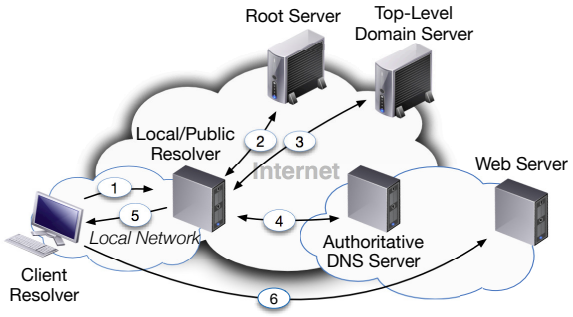


Fig. 1. DNS Resolution Process (Iterative query).

the lowest frequency to change their deployment patterns.

- Among the studied patterns (i.e., private, upstream, and hybrid), we observe that upstream achieves the highest performance while hybrid has the highest availability.
- We quantify the usage of ADNSes for cloud-hosting subdomains. We observe a noticeable growth on the usage of cloud-providing DNS service.

The remainder of this paper is organized as follows. We introduce DNS and ADNS deployment in §2. We describe the data sets used and our analysis methods in §3. We present the measurement results and analysis of ADNS deployment for top-ranking websites in §4. We profile the usage of ADNSes for cloud-hosting subdomains in §5. We survey related work in §6, and finally conclude the paper in §7.

II. Background

In this section, we give an overview of DNS and present the authoritative DNS deployment patterns for modern web services. In addition, we specially discuss the DNS deployment of cloud-hosting subdomains.

A. DNS Overview

Figure 1 shows the DNS components and the process of name resolution. A resolution routine on the client-end host, called *stub resolver*, issues a DNS lookup to a *recursive resolver*, a local DNS server deployed by the client's local network or a public DNS service [8], [10] located in a wide area network. Without considering the cache effects on the resolvers and intermediate servers, the recursive resolver will first contact the root server. The root server directs the resolver to query a top-level-domain (TLD) server (e.g., the .com TLD server). Similarly, the TLD server responds the resolver's query with the address of the *authoritative DNS* (ADNS) server for the corresponding domain. Next, the resolver queries the ADNS server for the address of the domain host, and finally the client can reach the Internet service as the recursive resolver returns the answer for name resolution.

B. ADNS Deployment Patterns

Figure 2 illustrates the steps of a client accessing the web services under three different ADNS deployments:

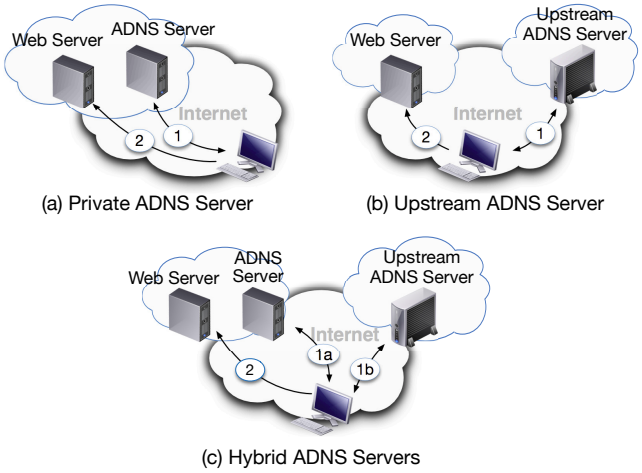


Fig. 2. ADNS Deployment for Web Services.

TABLE I
ADNS Deployment of Top 15 Sites[3]

Domain	ADNS
google.com	google.com
facebook.com	facebook.com
youtube.com	google.com
yahoo.com	yahoo.com
baidu.com	baidu.com
wikipedia.org	wikimedia.org
amazon.com	dynect.net, ultradns[4]
twitter.com	dynect.net
taobao.com	taobao.com
qq.com	qq.com
google.co.in	google.com
live.com	msft.net
sina.com.cn	sina.com.cn
linkedin.com	dynect.net, linkedin.com
weibo.com	sina.com.cn

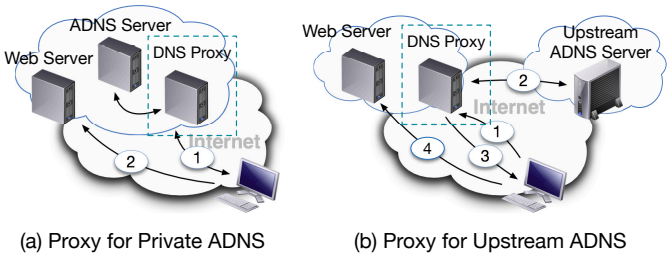
- **Private ADNS server:** The web service owners deploy their private authoritative DNS servers only within their own domains.[1]
- **Upstream ADNS server:** The web service owners delegate their authoritative name resolution to the upstream DNS-hosting service providers.[2]
- **Hybrid ADNS deployment:** The web service owners employ both the private DNS servers and the upstream ADNS servers for their authoritative name resolution.

[1]The domains hosting web services and private nameservers may also be located inside IaaS clouds. In such a case, the service provider runs the ADNS servers with cloud instances.

[2]We only consider the ADNS-hosting domains to identify the deployment, regardless of whether a website itself is hosted in private infrastructure or web-hosting companies.

[3]The ranking is from April 2015.

[4]The TLDs of Ultradns serving for `amazon.com` include `.net`, `.org`, `.info`, and `.co.uk`. Although Amazon offers a public DNS-hosting service (Route 53 [4]) for its cloud tenants, it delegates its DNS resolution to upstream providers. We infer that it is a historical reason: Amazon has been running the upstream ADNS for `amazon.com` since its establishment in 1995 and did not switch to private servers when expanding its business to cloud services.



(a) Proxy for Private ADNS (b) Proxy for Upstream ADNS

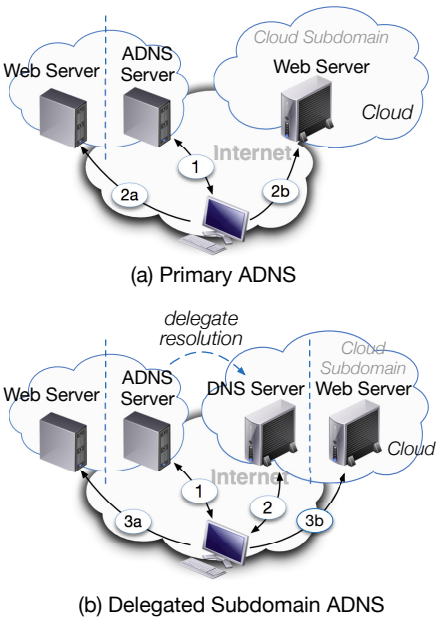
Fig. 3. DNS Proxy.

Table I lists the domains hosting authoritative DNS servers for the top 15 websites on Alexa's list [2]. Most of these top websites host the ADNS servers within their own domains since they are capable of maintaining a secure and reliable DNS infrastructure. However, `amazon.com` and `twitter.com` delegate their name resolution services to upstream ADNS providers, for which `amazon.com` uses two different vendors. `Linkedin.com` hosts ADNS servers in both its own domain and the upstream provider. Note that many top websites enable their primary ADNS servers to resolve the names for other domains they possess, such as Google for `youtube.com`, Microsoft for `live.com`, and Sina for `weibo.com`.

Use of DNS Proxy. To validate DNS traffic and protect the ADNS servers, the service owners may deploy the DNS proxy servers to control incoming[5] DNS queries and enable flexible management. Figure 3 shows the DNS proxy deployment for private and upstream ADNS servers. In both scenarios, the clients first contact the DNS proxy servers, and then the proxy servers transparently relay the queries to ADNS servers and return the answers back to clients. To date the DNS proxy servers have been developed as functionality-rich systems, such as Global Traffic Manager (GTM) [7] or Global Server Load Balancing (GSLB) [9], to optimize access performance and secure DNS servers. All these environments are recognized as the DNS proxy infrastructure in our study.

DNS for Cloud-hosting Subdomains. Figure 4 shows the steps for accessing the (partly) cloud-hosting subdomains. These subdomains are the partial sections of primary websites, hosted inside a cloud for achieving scalable infrastructure and providing extended services. The primary ADNS servers, deployed by either private servers or upstream services, could be used to direct users to the cloud subdomain, as shown in Figure 4(a). Also, some providers delegate the name resolution for cloud subdomains to a dedicated subdomain DNS server, as shown in Figure 4(b). The delegated DNS server could be deployed by (1) DNS software running within cloud instances, (2) DNS resolution service offered by the cloud provider, or

[5]The DNS proxy can also be used to control the outgoing DNS traffic in local networks. In addition, a specified re-routing service, called *Smart DNS proxy*, directs the users to access region-restricted or blocked content. In this paper, we only consider the DNS proxy that serves for incoming connections of ADNS servers.



(a) Primary ADNS

(b) Delegated Subdomain ADNS

Fig. 4. DNS Deployment for (Partly) Cloud-hosting Subdomains.

(3) the third-party upstream services. We examine the DNS deployment details for such cloud-hosting subdomains in §5.

III. Datasets and Methodology

This section introduces the datasets used in the study and presents our approaches to examine the deployment patterns of ADNS servers.

A. Datasets

Alexa's list [2] of the top 1-million websites is updated daily and based on the one-month average traffic. The list used in our study was downloaded in June 2014.

1) Authoritative DNS servers dataset: We collected the authoritative DNS server information for the top 1-million websites on Alexa's list through `dig` utility. Since the personal pages from several popular web services are individually included in Alexa's list and their hosting domains have been ranked, we eliminated those page links from our data with (1) a slash "/" after domain names, which typically indicates the sub-webpages, such as YouTube's user home pages `youtube.com/[user]`, and (2) the domains having the lower-level names before the second-level names[6], such as Blogspot's personal blog pages `[user].blogspot.com`. Finally, we collected the authoritative DNS information for 942,467 websites with 2,339,345 domain servers, consisting of 352,022 distinct nameservers.

Upon the dataset, we study the characteristics, such as performance and availability, of different ADNS deployment patterns by probing the collected servers. To eliminate the cache effect at intermediate servers, we issue the DNS queries

[6]They are identified by `publicsuffix` [13], a parser implemented by recognizing the domain suffixes from the Public Suffix List [12].

to designated ADNS servers by using the `@global-server` option of `dig` utility.

2) Cloud-hosting subdomain list: The list of cloud-hosting subdomains is extracted from a prior dataset [5], which includes the subdomains associated with Alexa's list and hosted in Amazon EC2 and Windows Azure. We reproduced the DNS server list since we found that a large number of DNS records have been changed. We also found many subdomains had been migrated to other cloud providers, but their DNS information could still be used for our study.

B. Determining ADNS Deployment Patterns

We first construct the list of authoritative DNS servers of each web domain by extracting `NS` records from the responses of `dig` probes, and then attempt to determine the ADNS deployment pattern. However, without a complete list of global upstream DNS providers to identify all DNS-hosting domains, it is impossible to have an automated method to accurately determine the pattern for every site. To ensure our study's effectiveness and accuracy, we design a heuristic method to capture *as many websites as possible for each pattern* by discarding the records likely to be miscategorized:

Step 1: First we extract the second-level name from each domain (e.g., `google` from `google.com`), and perform a substring search for that name in its every nameserver's hostname. Here we exclude the domains with one or two characters of second-level names since they may *coincidently* match the hostnames (e.g., the second-level name "`t`" from `t.co` matches the servers at `dynect.net`). We then move these domains into a `short-name` list to be determined later (Step 5).

Step 2: For each domain, we assign two variables, t_m and t_n, to record the number of matching and non-matching occurrences, respectively.

- If a certain server contributes t_m, we consider that it serves as a private nameserver located in the same or related domain with its web service. We collect those servers in a `private` list.
- For the servers without matching occurrence, we put them to an `upstream` list.
- For a domain with $t_m > 0$ and $t_n = 0$, we consider all its ADNS servers to be located in a related domain hosting its web service, and thus we refer to the domain as the private ADNS deployment.

For example, four nameservers `ns[1-4].google.com` serve for `google.com`. The second-level name `google` matches for all nameservers, which gives $t_m = 4$ and $t_n = 0$. Also, upon the general naming customs, this matching process has been able to recognize many domains that deploy private ADNS servers in separated and dedicated domains (e.g., `ebay.com` deploys its DNS servers in `ebaydns.com`).

However, some separated DNS-hosting domains cannot be identified from the simple substring search in this step (e.g., as shown in Table 1, the nameservers hosted in `sina.com.cn` serve for `weibo.com`).

Step 3: Our basic idea, to recognize such deployment, is to determine the deployment patterns of websites by the categories of their nameservers. Therefore, we would filter the `private` and `upstream` list (in this and the next step, respectively) to exclude the nameservers that might be miscategorized.

Despite the successful matching in step 1, the `private` list still includes some servers used as upstream services. For example, the nameserver `ns1.dnsmadeeasy.com` matches the domain `dnsmadeeasy.com`, and thus is recognized as a private DNS server. However, it also serves as an ADNS-hosting server for thousands of other websites.[7] Since the nameservers will be used to determine deployment patterns, we would eliminate such records from the `private` list.

In doing so, we first extract the domain part of each ADNS server by using the `publicsuffix` [13] parser (e.g., getting the domain part `bbc.co.uk` from the server `ns1.tcams.bbc.co.uk`). We then calculate the number of domains that each extracted domain part serves for, except for the domains with matching occurrences. For example, the ADNS servers in `ebaydns.com` support 64 domains in the dataset, 49 of which match the hostnames of servers (i.e., the second-level name `ebay` matches the domain `ebaydns.com`). This indicates the ADNS servers in `ebaydns.com` serve for 15 domains without matching occurrences. On the other hand, we find 8,487 domains supported by the servers located inside `dnsmadeeasy.com`.

The heuristic to filter the `private` list is from a general observation: *the number of domains served by private DNS servers would be less than the number of domains supported by the third-party upstream servers*. Since what we want is to have a private server list with accurate classification, simply discarding the servers with a certain number of served domains will be enough for our study.

Figure 5 demonstrates the cumulative distribution for the number of domains served by nameservers with *matching* occurrences. Not surprisingly, most of servers only support a few domains: about 96.7% of DNS-hosting domains serve for fewer than 20 web domains. We believe that it is safe to discard the servers administrating more than 20 domains in the `private` list, since (1) the matching process has excluded the majority of upstream servers, (2) most upstream service providers should have more than 20 served websites in such a large-scale dataset, and (3) while several errors exist, they would not have a significant effect on our study because the number of miscategorized websites is marginal ($< 20 \times the\ number\ of\ errors$).

To inspect the accuracy of our heuristic method, we extract and examine the results for the top 100 most popular

[7] In our study, the pattern of servers is determined by their specific role for an individual website. That is, in this example, `ns1.dnsmadeeasy.com` is a private DNS server for `dnsmadeeasy.com`, but an upstream server for other domains using its DNS-hosting service.

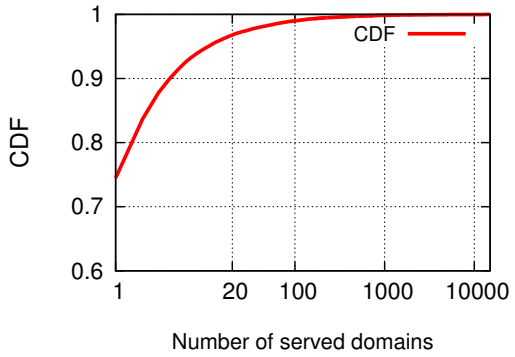


Fig. 5. CDF for the number of domains supported by individual DNS server under matching occurrence (*Note that the Y axis starts from 0.6*).

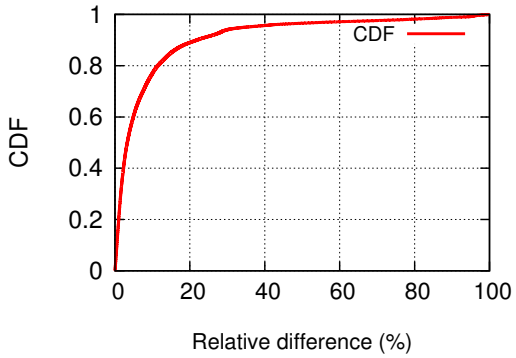


Fig. 6. CDF for the difference of responses by *ping* and *dig*.

websites. We find that the nameservers from three private domains (`google.com`, `facebook.com`, and `msft.net`) are removed from the `private` list since they serve for more than 20 domains. We argue that it also does not affect our study because (1) the ADNS deployment patterns of their primary websites have been identified as private due to the matching occurrence in step 2, and (2) only a small portion of the domains served by these private nameservers in the three domains above are discarded but most of them have the same ADNS deployment as their primary sites.[8]

According to the filtered `private` list, if any server in the `upstream` list also appears in the `private` list, we consider that it is deployed as a private server for the corresponding domain. We then move this server into the `private` list.

Step 4: We extract all domain parts from the servers in the `upstream` list and calculate how many websites those DNS-hosting domains support. For similar considerations with step 3, we discard the servers with fewer than 20 administrated domains since they could be private servers. We also will not move them to the `private` list because without any matching occurrence, we cannot ensure that they are private servers.

Step 5: We then update t_m and t_n for each unidentified domain (i.e., a domain not labeled in step 2 and in the `short-name` list) by examining its nameservers:

- if one nameserver appears in the `private` list, increase t_m by one;
- if one nameserver appears in the `upstream` list, increase t_n by one.

We finally determine the ADNS pattern as follows:

- $t_m > 0$ and $t_n = 0$: *private ADNS*
- $t_m = 0$ and $t_n > 0$: *upstream ADNS*
- $t_m > 0$ and $t_n > 0$: *hybrid ADNS*

As an example, we find six nameservers for `hao123.com`, a website-directory service provided by `baidu.com`. One nameserver is located in `hao123.com` and the other five are in `baidu.com`. Thereby, the matching step gives $t_m = 1$ and $t_n = 5$. In addition, the five nameservers hosted in `baidu.com` appear in the `private` list since they are also serving `baidu.com`. Finally, we have $t_m = 6$ and $t_n = 0$, and refer to this site as a private ADNS deployment.

We eliminate the domains whose nameservers cannot be identified by the `private` or `upstream` lists. Finally, we discard 19,956 (2.1%) records from the original dataset, of which 922,489 websites remain to be studied in our measurement. We understand our heuristics, filtering-based method may still not be perfect, but we believe that the method's accuracy is high enough for performing a large-scale measurement, with few errors at an acceptable level.

C. Validating the ADNS proxy

The use of the DNS proxy conceals the ADNS infrastructure and the proxy discovery is a challenging problem since the DNS proxy exhibits the same behavior as a private ADNS server from the viewpoint of external clients. We perform two different probing tests to study the proxy infrastructure.

We first discover the usage of a proxy by roughly estimating the DNS response latency within authoritative resolving infrastructure: if a DNS query undergoes a distinctly longer response delay than a *ping* probe, the website has a very high probability of using a proxy. We probe 138,240 private servers for 70,502 domains from three vantage points (at eastern-, middle-, and western-US), and we observe that 73% of servers respond to the *ping* probes. Then, for each of those servers, we attempt to identify and normalize the time difference in response latency between *ping* probes and DNS queries. The normalized difference in response latency (i.e., round-trip-time, *RTT*) between DNS and *ping* is computed as below:

$$(RTT_{\mathbf{dns}} - RTT_{\mathbf{ping}})/RTT_{\mathbf{dns}}$$

We measure the normalized difference values in three vantage points and define their average as the relative difference for each server. Figure 6 plots the distribution of relative difference. We observe that (1) the majority of servers (close 90%) whose relative difference values are less than 20%, but (2) indeed there are 3.4% and 1.8% of servers whose relative difference values are higher than 50% and 80%, respectively, i.e., their DNS queries have response times 50% and 80%

[8]The nameservers of these sites are moved back to the `private` list in the following steps.

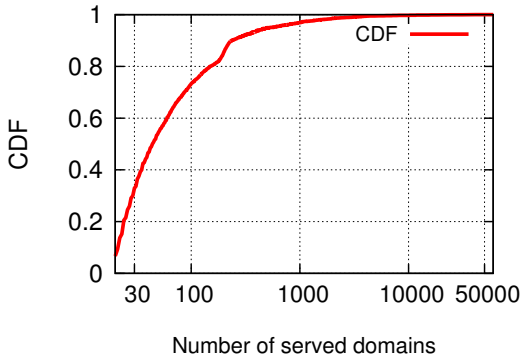


Fig. 7. CDF for served domains of DNS-hosting domains.

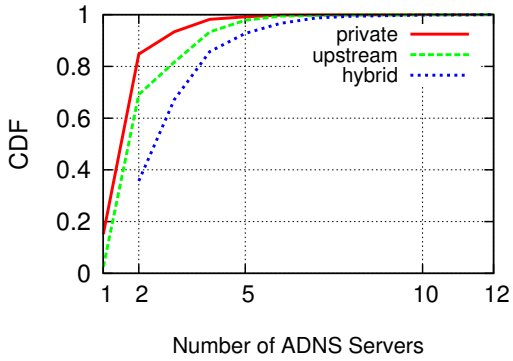


Fig. 8. CDF for the number of ADNS servers.

TABLE II
SUMMARY OF DEPLOYMENT PATTERNS

Patterns	Top 100	Top 1k	Top 10k	All	
	%	%	%	#	%
Private	68	34.1	18.76	70,520	7.6
Upstream	27	60.2	77.17	838,605	91.0
Hybrid	5	5.7	4.07	13,364	1.4
Total	100	100	100	922,489	100

TABLE III
TOP 10 SITES FOR EACH DEPLOYMENT PATTERN

Private		Upstream		Hybrid	
rk.	Domain	rk.	Domain	rk.	Domain
1	google.com	9	amazon.com	15	linkedin.com
2	facebook.com	12	twitter.com	20	ebay.com
3	youtube.com	30	pinterest.com	65	cnn.com
4	yahoo.com	34	tumblr.com	84	ebay.de
5	baidu.com	38	paypal.com	99	ebay.co.uk
6	wikipedia.org	39	instagram.com	115	nytimes.com
7	qq.com	42	xvideos.com	120	pixnet.net
8	taobao.com	45	imdb.com	148	livedoor.com
10	live.com	48	ifeng.com	167	skype.com
11	sina.com.cn	49	amazon.co.jp	171	ups.com

higher than the RTTs of *ping* probes. Assuming the same or at least similar routing paths for consecutive *ping* probes and DNS queries, even if we cannot accurately identify the proxy for each website, our probing results clearly indicate that the use of ADNS proxy infrastructure does exist but the usage is very limited.

If a proxy enables the cache, we cannot use the relative difference to detect the use of a DNS proxy anymore, since the cache will respond to the queries and no significant additional delay would be noticed. However, different from an ADNS direct response, in which TTL is a fixed default value, the TTL value of a proxy cached record decreases with time elapse. This is because the cache at a DNS proxy will age the cached records like a local resolver. That is, the answers from a proxy will have reduced TTLs. Moreover, the time resolution in TTL is in seconds. Therefore, we send two successive queries with an interval of 10s to these ADNS addresses and then use the reduced TTL values to detect the existence of a DNS proxy. In other words, if we can detect that the difference of TTLs between two successive DNS queries is round 10s, a proxy infrastructure may have been adopted to conceal the actual ADNS servers. However, by probing 70,520 domains with private deployment, we only find 75 (0.001%) domains where TTL-reduced records from their ADNS servers occur. This observation is consistent with the proxy detection result above, i.e., the ADNS proxy infrastructure is not used by most service owners. Furthermore, comparing the two results (3.4% vs 0.001%), we can see that most ADNS proxy servers simply relay the authoritative records to clients and no cache is used.

IV. MEASUREMENT RESULTS

This section presents the measurement results and analysis for the websites on Alexa's list. First we quantify the usage of deployment patterns and examine the fundamental deployment configurations, such as the number of servers and TTLs of DNS records. Then we present the performance study based on the probes from the geo-distributed locations and analyze the availability for different patterns.

A. Overview

1) Deployment Patterns: Table II summarizes the pattern recognition in our dataset, based on the breakdown by the top 100, 1,000, 10,000, and all records. The majority of websites delegate their authoritative name resolution to upstream DNS-hosting services for the simple management. The high-ranking sites are much more likely to deploy their own ADNS servers: the fraction of private deployment decreases sharply as the number of studied domains increases. In addition, the percentage of hybrid-deploying domains remains stable on about 4-6% within top 10,000 websites but overall only 1.4% of websites are recognized as such a deployment. Table III lists the top 10 web domains of each deployment pattern. Many emerging top websites, especially the social sites popular on mobile web, such as Twitter, Pinterest, and Tumblr, use the upstream services to facilitate the quick and convenient deployment.

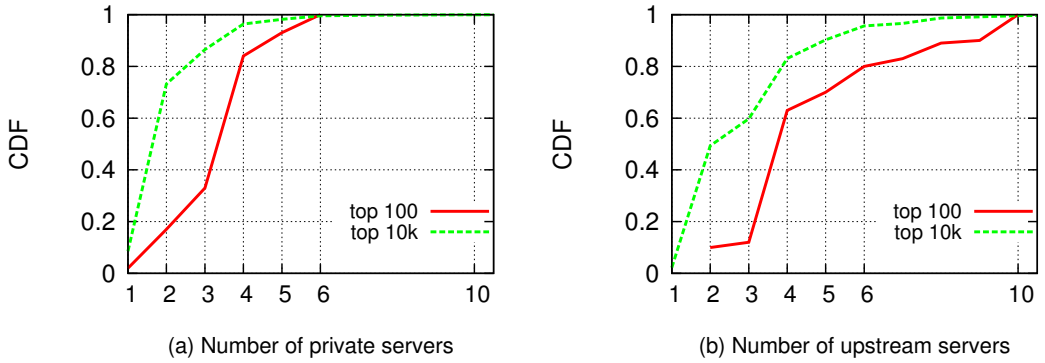


Fig. 9. CDFs for the number of ADNS servers of top sites (private and upstream).

TABLE IV
TOP DNS-HOSTING DOMAINS

Domain	# of servers	# of websites
domaincontrol.com	96	61,455
cloudflare.com	107	27,821
hostgator.com	4,913	22,483
ovh.net	1,960	14,004
bluehost.com	6	12,307
worldnic.com	100	10,855
dnsmadeeasy.com	15	8,487
dnspod.net	12	8,404
name-services.com	13	8,338
dreamhost.com	5	7,593

2) DNS-hosting Providers: Here we examine the upstream DNS-hosting providers profiled by our dataset. Figure 7 shows the distribution of the number of served domains for extracted DNS-hosting domains. The popularity of upstream domains is quite skewed. We observe that about 70% of hosting domains support fewer than 100 websites, and only 1.4% of hosting domains serve websites more than 1,000.

Table IV lists the top 10 DNS-hosting domains in terms of the number of websites they support in Alexa's list, with the number of servers identified in our dataset. These top 10 domains serve for 26.7% of websites using upstream ADNS. If we include all the domains with more than 1,000 served sites (i.e., 1.4% of hosting domains), this proportion increases to 62.6%. Meanwhile, as the second column of Table IV shows, the quantities of DNS-hosting servers vary considerably,[9] from a few to thousands. This implies that the DNS-hosting providers employ different system designs and implementations to achieve the load-balancing and reliable upstream ADNS services.

Note that the numbers of websites in Table IV are produced by extracting hosting domains, not the service providers. In fact, several providers offer hosting services through multiple domains, e.g., Ultradns (see footnote 4), and Amazon's DNS service. Amazon deploys a set of hosting domains (256 found), named `awsdns-××.com/net/org/co.uk`, among which the quantities of served web domains remain balanced and stable (220-320).

B. Number of Nameservers

We quantify the usage of ADNS servers in terms of the deployment patterns. Figure 8 plots the cumulative distribution of the number of ADNS servers for each pattern.[10]

Generally, the websites using private ADNS servers tend to deploy slightly fewer servers than the sites using upstream services. This indicates that those sites have the ability to maintain a reliable DNS infrastructure, and various back-end techniques may be used, such as load balancing and failover. On the other hand, simply using more upstream servers is a convenient practice to achieve reliable name resolution. In addition, when looking at the quantities in terms of ranks, as shown in Figure 9, we observe that the high-ranked sites have more servers than others to handle the high-volume accesses. Figure 10 shows the CDFs for the numbers of private and upstream servers in hybrid deployment. They exhibit very similar distributions, and the number of servers for either type is less than that in pure private or upstream pattern.

In our dataset, there are only 209 ($< 0.02\%$) and 28 ($< 0.003\%$) domains that have more than 12 and 13 DNS servers, respectively. For easy presentation, we do not show those domains in Figures 8-10. By briefly examining those websites equipped with a large number of ADNS servers, we recognize that 60% of them having more nameservers are mainly for the purpose of *mutual delegation*. That is, a group of websites are served by a set of nameservers that consist of the private servers from every site. These websites with mutual delegation mainly fall into two categories: porn-related and loan-related websites, perhaps for circumventing the local laws or web inspection.

C. Time-to-Live Values

We now profile the Time-to-Live (TTL) settings of ADNS servers. Figure 11(a) shows the cumulative distribution of

[9]These servers are identified by their hostnames. In some cases, the service providers prefix the clients' domains to their nameservers to form the client-specific hostnames of ADNSes. This causes the over-estimation of the number of servers for some providers (but not a common case).

[10]We omit the plotting for the overall result since it would show a similar distribution with the upstream pattern due to its dominant quantity.

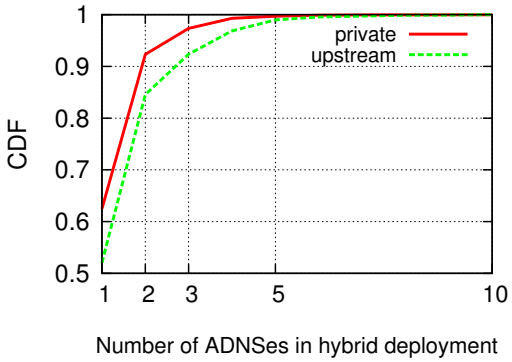


Fig. 10. CDF for the number of private and upstream servers in hybrid deployment.

TTL values of NS records for different ADNS deployment patterns. We observe that the NS records extracted from the nameservers of a hybrid deployment have shorter TTL values than the records of private and upstream patterns. One possible reason to explain why this happens is that: the domain that decides to deploy hybrid DNS servers demands high availability and reliability, thus its system administrators are more meticulous and tend to maintain resilient configuration to quickly respond to the changes on ADNS settings.

Figure 11(b) shows the distribution of TTL values of A records. The A records of websites using upstream services have shorter TTL values since the majority of those websites run small businesses and may change their service deployment more frequently. The larger TTL values of websites following the hybrid pattern imply that those websites indeed would like to maintain more stable services.

The larger TTL values of NS records have been identified by many prior studies [26], [28]. However, when comparing Figures 11(a) and 11(b), we observe that the hybrid deployment pattern has very similar TTL distributions for the NS and A records, while the NS records of private and upstream servers indeed demonstrate the larger TTL values than their A records.

D. Life-cycle of ADNS Servers

Our probing experiments are performed based on the ADNS server dataset collected in June 2014. After that, we also reproduced two additional lists of ADNS server information in December 2014 and April 2015, for the same Alexa's list in June 2014, to examine how many records have been changed. We observe that 20.4% of ADNS records have been changed in December 2014, and 33.5% of records have been changed in April 2015.[11]

We study the ADNS changes at different levels. We refer to all changes on ADNS records as *server change*, i.e., the nameservers to be added, removed, or relocated. The *domain change* means that the nameservers are relocated to different hosting domains. The *pattern change* means that the websites re-deploy their ADNSes with a different infrastructure. Clearly, the domain change indicates the server change, and the pattern change indicates both the server change and the domain change.

[11]All the change rates in April 2015 are based on the comparison with the original dataset in June 2014.

TABLE V
SUMMARY OF ADNS CHANGES (%)

Patterns	Dec. 2014			Apr. 2015		
	Ser.	Dom.	Pat.	Ser.	Dom.	Pat.
Private	23.2	4.57	2.90	37.5	7.09	4.51
Upstream	16.7	8.56	1.78	24.7	13.59	2.47
Hybrid	25.3	4.35	3.22	37.2	6.66	4.67
Overall	20.4	6.34	2.53	33.5	8.05	3.68

The detailed breakdown appears in Table V. Although a large number of ADNS records have been changed, we observe that the deployment patterns remain stable. The websites using upstream DNS services frequently change their hosting domains (i.e., third-party service providers), due to the low cost to migrate their authoritative records; but they also have the lowest frequency to change their deployment patterns, due to the high cost to deploy extra infrastructures to host their ADNS records.

E. Performance

To evaluate the DNS performance, we performed the DNS lookups for each nameserver of every domain, which involves 2,223,972 nameservers for 922,489 websites, from 70 globally distributed PlanetLab [11] nodes, and then we clustered the response times into groups according to the deployment patterns. The probings are performed twice each week during a one-month period. Figure 12 plots the cumulative distribution of the response times for each pattern. It demonstrates that the upstream services have a small but noticeable performance advantage, since the mainstream DNS-hosting providers have established the distributed infrastructure and spent significant efforts in optimizing their global access.

F. Availability

To examine the availability of authoritative DNS servers, we first analyze the responsiveness from the the probing experiments above. During each probing period, we record a query's response time and retry the unresponsive servers up to three times before we drop the probes. We then calculate the average rates of successful probes. We refer to the results from the performance experiment as Probe 1.

However, the performance probes focus on the response time and do not reflect the availability for a certain period of time. Thus, besides conducting the performance probes, we also perform an active probe experiment with an exponentially distributed interval of a one-hour mean, the same method used in [32], where the availability is defined as the ratio of the number of probes being responded and the number of probes being sent. We select four geo-distributed nodes (located at eastern- and western-US, Europe, and Asia) to issue the probes

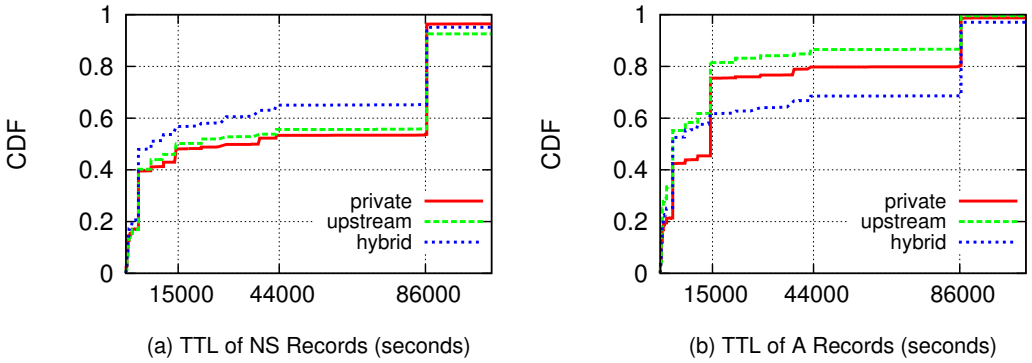


Fig. 11. CDFs for TTL values.

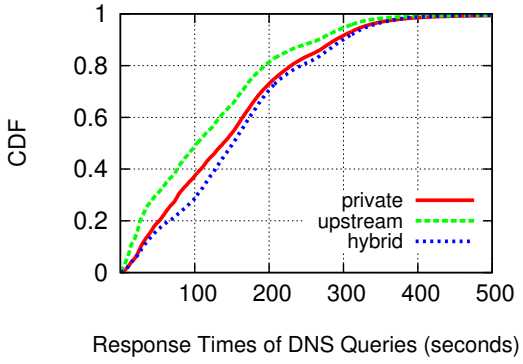


Fig. 12. CDF for Response Times of DNS Queries.

TABLE VI
SUMMARY OF AVAILABILITY (%)

Patterns	Probe 1		Probe 2	
	Server	Domain	Server	Domain
Private	92.54	96.77	95.63	98.65
Upstream	94.44	97.76	98.87	99.04
Hybrid	94.33	97.71	96.93	99.89
Overall	93.54	97.47	97.39	98.98

and run the measurement for approximately five weeks. We refer to the results from the exponential probes as Probe 2.

Table VI summarizes the availability statistics from both probe sets. For each deployment pattern, its server availability refers to the ratio between the successfully responded nameservers and the probed nameservers. The domain availability indicates the chance in percentage that at least one of the domain's nameservers responds to the probes. The majority of ADNS servers are available for almost the entire duration. In both probe sets, the servers in upstream deployment have the highest availability since they are more powerful and expensive servers aiming to provide DNS services for many customers. At the domain level, as one would expect, the hybrid deployment indeed exhibits higher availability than the other two deployment patterns in Probe 2.[12]

We observe that a few servers become dead during the experiment of Probe 2. We consider a nameserver becoming dead if the server remains no response for at least one week and until the end of the experiment. The dead servers/domains are removed from the results in Table VI. We observe that 3.3% of servers become dead, 87.4% of which are private servers. We further perform another probing in April 2015 to check the status of those severs that have been identified as dead in Probe 2. We find that only 6.4% of them respond to the probes, which implies that the corresponding domains having responsive servers now may have re-deployed their ADNS servers.

TABLE VII
SUMMARY OF DEPLOYMENT PATTERNS FOR CLOUD SUBDOMAINS (%)

Patterns	%
private	11.49
cloud-private	5.40
upstream	15.20
cloud-provided	66.59
hybrid	1.32

V. CLOUD-HOSTING SUBDOMAINS

In this section, we explore the DNS deployment for subdomains hosted in IaaS clouds.

A. Deployment

We first extract the cloud-hosting subdomain list from [5] and reproduce the ADNS servers of subdomains in July 2014. Since the subdomains are associated with the websites on Alexa's list,[13] we use our `private` list to identify the private nameservers. Also, due to the simplicity of cloud DNS and dominant quantity of subdomains in Amazon EC2, we only examine the EC2-using subdomains in our study.

[12]In Probe 1, the upstream and hybrid deployment patterns demonstrate similar availability at the domain level, perhaps due to the occurrence of temporary network outage or congestion during the probes.

[13]The subdomain list [5] was generated by Alexa's list in February 2013. We exclude the subdomains whose primary domains are not with Alexa's list.

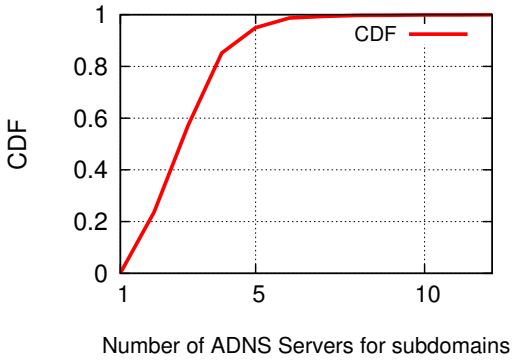


Fig. 13. CDF for the number of ADNS servers.

Here we split the upstream service providers into two categories: the *cloud-provided* DNS and *non-cloud-provided* services (we still call them upstream providers). The ADNS-hosting servers of Amazon are simply identified by their host-names, including `amazonaws`, `awsdns`, and `CloudFront`. We also add a new category of ADNS deployments, called *cloud-private*, to represent the private DNS servers running atop EC2 instances. We extract the primary domain parts and perform the search like our previous step in Section 3. If we recognize a matching occurrence and the nameservers are not in the `private` list, we check if its IP address is located in Amazon EC2's IP range [3].

Table VII summarizes the usage of the ADNS deployment patterns, showing a different pattern usage. However, it would be similar to Table II if we combine the private and cloud-private, as well as the upstream and cloud-provided, into one category. We also observe a usage growth of cloud-provided DNS deployment from the original dataset (about 54%). Figure 13 plots the number of nameservers used by the cloud-hosting subdomains, which is slightly higher than the numbers shown in Figure 8.

B. Subdomain Delegation

Recall that in Figure 4(b), the primary websites may delegate the resolution of their cloud-hosting subdomains to dedicated nameservers. We identify the delegation by extracting all subdomains' CNAME records and search the CNAMEs in all NS records. We observe that 4.7% of cloud-hosting subdomains use this deployment style, and 97.6% of them delegate the subdomains to upstream DNS-hosting providers. Not surprisingly, 89.4% of upstream servers are identified as cloud-provided nameservers.

C. Life-cycle of Servers

Table VIII summarizes the statistics of the changes of DNS records for cloud-hosting subdomains. We recollected the DNS information in April 2015 to compare this dataset with the original one [5] (from February 2013) and the dataset used in our study (from July 2014). We observe a much greater change rate for both nameservers and hosting domains than the identified changes in Table V.

TABLE VIII
SUMMARY OF DNS CHANGES FOR CLOUD SUBDOMAINS (%)

Change Pattern	Jul. 2014	Apr. 2015
Server	35.49	47.51
Domain	18.27	24.68
Pattern	14.64	20.39

VI. Related Work

Pang *et al.* [32] presented a comprehensive DNS study by characterizing the properties of local and authoritative DNS infrastructures for the availability, usage, and deployment of DNS. Sisson [15] presented a survey that reports the number of DNS servers on the Internet and various aspects on configuration, such as the recursive support and security configuration. Our work revisits several of their key findings and examines these properties with respect to the evolution of ADNS deployment associated with the use of cloud and upstream resolvers. Schomp *et al.* [35] presented the measurement techniques to discover the client-side DNS infrastructure and studied its behavior on caching. He *et al.* [27] examined how modern web services are using the cloud for deployment of their front ends. Our work focuses on the deployment and characteristics of authoritative DNS infrastructure for cloud-hosting services.

There have been studies to investigate the DNS infrastructure. Gao *et al.* [26] conducted a comprehensive measurement study on global DNS resolvers to reaffirm some findings in previous works and reveal the key differences from root and local perspectives, respectively. Callahan *et al.* [21] passively monitored DNS traffic within a residential network to understand server behaviors and properties of the modern DNS system, such as DNS responses and the violation of TTLs. Jung *et al.* [28] presented a detailed analysis of DNS traces to evaluate the client-perceived performance and the effectiveness of DNS caching, and to simulate the effect of varying TTLs and sharing caches. Liang *et al.* [29] investigated the latency of upper DNS hierarchy and studied the impact of uneven distribution of top-level DNS servers on end-user latency. Pappas *et al.* [33] studied the reduced availability and increased query delays caused by DNS misconfigurations, and presented three specific widespread types of misconfigurations: lame delegation, diminished server redundancy, and cyclic zone dependency. Ager *et al.* [18] compared the local DNS resolvers against open DNS resolvers, i.e., GoogleDNS and OpenDNS, to examine the latency of DNS resolvers and the content of DNS caches.

There have also been prior works to examine the characteristics of DNS services from various aspects. Liston *et al.* [30] identified the diversity of DNS performance and investigated the degree to which they vary from site to site. Deccio *et al.* [25] proposed a model for server dependencies to measure DNS availability. Castro *et al.* [22] characterized the workload at root servers and analyzed some trends for DNS evolution, such as DNSSEC and DNS IPv6. Berger *et al.* [19] examined the associations between IPv6 addresses and IPv4

addresses of Internet DNS resolvers. Ramasubramanian *et al.* [34] studied the security aspects of nameserver dependencies and delegations. Cranor *et al.* [23] identified the distribution of DNS servers in clusters. Otto *et al.* [31] studied the end-to-end impact of using remote DNS services in CDN, which breaks the assumption that the location of clients' DNS resolvers is close to the actual location of clients.

VII. Conclusion

In this paper, we conduct a large-scale measurement study to quantify the deployment patterns of authoritative DNS servers and examine the characteristics of the patterns. We develop a simple heuristics-based method to determine the ADNS deployment patterns of web domains on Alexa's top 1-million list. We observe that a majority of websites host the ADNSes in upstream services, but the top-ranked sites tend to deploy their own ADNS servers. We then perform a probing experiment and observe the performance advantage from upstream services. The hybrid pattern exhibits the high availability due to the backup and redundant deployment. Finally we examine the usage of ADNSes for cloud-hosting subdomains, and observe a noticeable growth in the use of cloud-providing DNS hosting services. In our future work, we will further investigate the usage of spam ADNSes, profile the deployment patterns of TLDs, study the impact of (multiple) CDNs and DNSSEC on different ADNS deployment patterns, and examine more details for cloud-hosting ADNSes.

Acknowledgments

We would like to thank the anonymous reviewers and our shepherd, Harsha V. Madhyastha, for their detailed and insightful comments, which help to improve the quality of this paper. This work was partially supported by ONR grant N00014-13-1-0088 and NSF CNS 1421747. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Akamai Enhanced DNS. http://www.akamai.com/html/solutions/enhanced_dns.html.

[2] Alexa Top Sites. http://www.alexa.com/topsites. The top 1-million list is available at http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[3] Amazon IP Range. http://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html.

[4] Amazon Route 53. http://aws.amazon.com/en/route53/.

[5] Cloud-using Subdomain List. *Data Sets for Cloud Measurement Project*, University of Wisconsin - Madison. http://pages.cs.wisc.edu/~keqhe/cloudmeasure_datasets.html.

[6] Dyn solutions. http://dyn.com/managed-dns/.

[7] Global Traffic Manager, *F5 Networks, Inc*. http://www.f5.com/pdf/products/big-ip-global-traffic-manager-ds.pdf.

[8] Google Public DNS. https://developers.google.com/speed/public-dns/.

[9] NetScaler Global Server Load Balancing, *Citrix Systems, Inc*. http://support.citrix.com/article/CTX123792.

[10] OpenDNS. https://www.opendns.com/.

[11] PlanetLab. https://www.planet-lab.org/.

[12] Public Suffix List, *Mozilla*. https://publicsuffix.org/list/.

[13] Publicsuffix, https://pypi.python.org/pypi/publicsuffix/.

[14] Ultradns. http://www.neustar.biz/services/dns-services.

[15] G. Sisson. DNS Survey: October 2010. http://dns.measurement-factory.com/surveys/201010/.

[16] State of the Cloud DNS, *CloudHarmony, Inc.* https://cloudharmony.com/reports/state-of-the-cloud-dns-report.

[17] Measuring the Health of the Domain Name System. In *Report of the 2nd Annual Global Symposium on DNS Security, Stability and Resiliency*. https://www.icann.org/en/system/files/files/dns-ssr-symposium-report-1-03feb10-en.pdf

[18] B. Ager, M. Wolfgang, G. Smaragdakis, and S. Uhlig. Comparing DNS Resolvers in the Wild. In *Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement* (IMC'10), pages 15-21, 2010.

[19] A. Berger, N. Weaver, R. Beverly, and L. Campbell. Internet Nameserver IPv4 and IPv6 Address Relationships. In *Proc. of the 2013 ACM SIGCOMM Conference on Internet Measurement* (IMC'13), pages 267-278, 2013.

[20] N. Brownlee, K. Claffy, and E. Nemeth. DNS Measurements at a Root Server. In *Proc. of IEEE Global Telecommunications Conference* (GLOBECOM), pages: 1672-1676 (vol.3), Nov. 2001.

[21] T. Callahan, M. Allman, and M. Rabinovich. On Modern DNS Behavior and Properties. *ACM SIGCOMM Computer Communication Review*, 43(3): 7-15, July 2013.

[22] S. Castro, M. Zhang, W. John, D. Wessels, and K. C. Claffy, Understanding and Preparing for DNS Evolution, In *Proc. of the 2nd International Workshop on Traffic Monitoring and Analysis* (TMA'10), pages 1-16, Zurich, Switzerland, 2010.

[23] C. D. Cranor, E. Gansner, B. Krishnamurthy, and O. Spatscheck. Characterizing Large DNS Traces Using Graphs. In *Proc. of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 55-67, 2001.

[24] P. B. Danzig, K. Obraczka, and A. Kumar. An Analysis of Wide-Area Name Server Traffic: A Study of the Internet Domain Name System. In *Proc. of the 1992 ACM SIGCOMM Conference*, pages 281-292, 1992.

[25] C. Deccio, J. Sedayao, K. Kant, and P. Mohapatra. Measuring Availability in the Domain Name System. In *Proc. of the 29th International Conference on Computer Communications* (INFOCOM'10), 2010.

[26] H. Gao, V. Yegneswaran, Y. Chen, P. Porras, S. Ghosh, J. Jiang, and H. Duan. An Empirical Reexamination of Global DNS Behavior. In *Proc. of the ACM SIGCOMM 2013 Conference*, pages 267-278, 2013.

[27] K. He, A. Fisher, L. Wang, A. Gember, A. Akella, and T. Ristenpart. Next Stop, the Cloud: Understanding Modern Web Service Deployment in EC2 and Azure. In *Proc. of the 2013 ACM SIGCOMM Conference on Internet Measurement* (IMC'13), pages 177-190, 2013.

[28] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, DNS Performance and the Effectiveness of Caching, *IEEE/ACM Transactions on Networking*, October 2002, Volume 10, Number 5.

[29] J. Liang, J. Jiang, H. Duan, K. Li, and J. Wu. Measuring Query Latency of Top Level DNS Servers. In *Proc. of the 14th International Conference on Passive and Active Measurement* (PAM'13), pages 145-154, 2013.

[30] R. Liston, S. Srinivasan, and E. Zegura. Diversity in DNS Performance Measures. In *Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, pages 19-31, 2002.

[31] J. S. Otto, A. S. Mario, J. P. Rula, and F.E. Bustamante. Content Delivery and the Natural Evolution of DNS: Remote DNS Trends, Performance Issues and Alternative Solutions. In *Proc. of the 2012 ACM SIGCOMM Conference on Internet Measurement* (IMC'12), pages 523-536, 2012.

[32] J. Pang, J. Hendricks, A. Akella, R. De Prisco, B. Maggs, and S. Seshan. Availability, Usage and Deployment Characteristics of the Domain Name System, In *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement* (IMC'04), pages 1-14, 2004.

[33] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang. Impact of Configuration Errors on DNS Robustness. In *Proc. of the 2004 ACM SIGCOMM Conference*, pages 319-330, 2004.

[34] V. Ramasubramanian and E. G. Sirer. Perils of Transitive Trust in the Domain Name System. In *Proc. of the 5th ACM SIGCOMM Conference on Internet Measurement* (IMC'05), pages 379-384, 2005.

[35] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman. On Measuring the Client-Side DNS Infrastructure, In *Proc. of the 2013 ACM SIGCOMM Conference on Internet Measurement* (IMC'13), pages 77-90, 2013.

[36] D. Wessels, M. Fomenkov, N. Brownlee, and K. C. Claffy. Measurements and Laboratory Simulations of the Upper DNS Hierarchy. In *Proc. of the 5th International Workshop on Passive and Active Measurement* (PAM'04), 2004.