Understanding and preparing for DNS evolution

Sebastian Castro[1,2], Min Zhang[1], Wolfgang John[1,3],
Duane Wessels[1,4] and kc claffy[1]
{secastro,mia,johnwolf,kc}@caida.org
wessels@dns-oarc.net

[1]CAIDA, University of California, San Diego
[2]NZRS, New Zealand
[3]Chalmers University of Technology, Sweden
[4]DNS-OARC

Abstract. The Domain Name System (DNS) is a crucial component of today's Internet. The top layer of the DNS hierarchy (the root nameservers) is facing dramatic changes: cryptographically signing the root zone with DNSSEC, deploying Internationalized Top-Level Domain (TLD) Names (IDNs), and addition of other new global Top Level Domains (TLDs). ICANN has stated plans to deploy all of these changes in the next year or two, and there is growing interest in measurement, testing, and provisioning for foreseen (or unforeseen) complications. We describe the *Day-in-the-Life* annual datasets available to characterize workload at the root servers, and we provide some analysis of the last several years of these datasets as a baseline for operational preparation, additional research, and informed policy. We confirm some trends from previous years, including the low fraction of clients (0.55% in 2009) still generating most misconfigured "pollution", which constitutes the vast majority of observed queries to the root servers. We present new results on security-related attributes of the client population: an increase in the prevalence of DNS source port randomization, a short-term measure to improve DNS security; and a surprising decreasing trend in the fraction of DNSSEC-capable clients. Our insights on IPv6 data are limited to the nodes who collected IPv6 traffic, which does show growth. These statistics serve as a baseline for the impending transition to DNSSEC. We also report lessons learned from our global trace collection experiments, including improvements to future measurements that will help answer critical questions in the evolving DNS landscape.

1 Introduction

The DNS is a fundamental component of today's Internet, mapping domain names used by people and their corresponding IP addresses. The data for this mapping is stored in a tree-structured distributed database where each nameserver is authoritative for a part of the naming tree. The *root nameservers* play a vital role providing authoritative referrals to nameservers for all top-level domains, which recursively determine referrals for all host names on the Internet,

among other infrastructure information. This top (root) layer of the DNS hierarchy is facing three dramatic changes: cryptographically signing the root zone with DNSSEC, deploying Internationalized Top-Level Domain (TLD) Names (IDNs), and addition of other new global Top Level Domains (TLDs). In addition, ICANN and the root zone operators must prepare for an expected increase in IPv6 glue records in the root zone due to the exhaustion of IPv4 addresses. ICANN currently plans to deploy all of these changes within a short time interval, and there is growing interest in measurement, testing, and provisioning for foreseen (or unforeseen) complications.

As part of its DNS research activities, in 2002 CAIDA responded to the Root Server System Advisory Committee's invitation to help DNS root operators study and improve the integrity of the root server system. Based on the few years of trust we had built with these operators, in 2006 we asked them to participate in a simultaneous collection of a day of traffic to (and in some cases from) the DNS root nameservers. We collaborated with the Internet Systems Consortium (ISC) and DNS Operation and Research Center (DNS-OARC) in coordinating four annual large-scale data collection events that took place in January 2006, January 2007, March 2008, and March 2009. While these measurements can be considered prototypes of a *Day in the Life of the Internet* [8], their original goal was to collect as complete a dataset as possible about the DNS root servers operations and evolution, particularly as they deployed new technology, such as anycast, with no rigorous way to evaluate its impacts in advance. As word of these experiments spread, the number and diversity of participants and datasets grew, as we describe in Section 2. In Section 3 we confirm the persistence of several phenomenon observed in previous years, establishing baseline characteristics of DNS root traffic and validating previous measurements and inferences, and offering new insights into the pollution at the roots. In Section 4 we focus on the state of deployment of two major security-related aspects of clients querying the root: source port randomization and DNSSEC capability. We extract some minor insights about IPv6 traffic in Section 5 before summarizing overall lessons learned in Section 6.

2 Data sets

On January 10–11, 2006, we coordinated concurrent measurements of three DNS root server anycast clouds (C, F, and K, see [13] for results and analysis). On January 9–10, 2007, four root servers (C, F, K, and M) participated in simultaneous capture of packet traces from almost all instances of their anycast clouds [5]. On March 18–19, 2008, operators of eight root servers (A, C, E, F, H, K, L, and M), five TLDs (.ORG, .UK, .BR, .SE, and .CL), two Regional Internet Registries (RIRs: APNIC and LACNIC), and seven operators of project AS112 joined this collaborative effort. Two Open Root Server Network (ORSN) servers, B in Vienna and M in Frankfurt, participated in our 2007 and 2008 collection experiments. On March 30–April 1, 2009, the same eight root servers participated in addition to seven TLDs (.BR, .CL, .CZ, .INFO, .NO, .SE, and .UK), three

RIRs (APNIC, ARIN, and LACNIC), and several other DNS operators [9]. To the best of our knowledge, these events deliver the largest simultaneous collection of full-payload packet traces from a core component of the global Internet infrastructure ever shared with academic researchers. DNS-OARC provides limited storage and compute power for researchers to analyze the DITL data, which for privacy reasons cannot leave OARC machines.[1] For this study we focus only on the root server DITL data and their implications for the imminent changes planned for the root zone.

Each year we gathered more than 24 hours of data so that we could select the 24-hour interval with the least packet loss or other trace damage. The table in Fig. 1 presents summary statistics of the most complete 24-hour intervals of the last three years of DITL root server traces. Figure 1 (right) visually depicts our data collection gaps for UDP (the default DNS transport protocol) and TCP queries to the roots for the last three years. The darker the vertical bar, the more data we had from that instance during that year. The noticeable gaps weaken our ability to compare across years, although some (especially smaller, local) instances may have not received any IPv6 or TCP traffic during the collection interval, i.e., it may not always be a data gap. The IPv6 data gaps were much worse, but we did obtain (inconsistently) IPv6 traces from instances of four root servers (F, H, K, M), all of which showed an increase of albeit low levels of IPv6 traffic over the 2-3 observation periods (see Section 5).

3 Trends in DNS workload characteristics

To discover the continental distribution of the clients of each root instances measured, we mapped the client IP addresses to their geographic location (continent) using NetAcuity [2]; the location of the root server instances is available at *www.root-servers.org* [1]. Not surprisingly, the 3 unicast root servers observed had worldwide usage, i.e., clients from all over the globe. Fifteen (15) out of the 19 observed global anycast instances also had globally distributed client populations (exceptions were f-pao1, c-mad1, k-delhi, m-icn[2]). Our observations confirm that anycast is effectively accomplishing its distributive goals, with 42 of the 46 local anycast instances measured serving primarily clients from the continent they are located in (exceptions were f-cdg1, k-frankfurt, k-helsinki, f-sjc1[3]). We suspect that the few unusual client distributions results from particular BGP routing policies, as reported in Liu et al.[13] and Gibbard [10].

Figure 3 shows fairly consistent and expected growth in mean query rates observed at participating root servers. The geographic distribution of these queries spans the globe, and similar to previous years [13] suggest that anycast at the root servers is performing effectively at distributing load across the now much more globally pervasive root infrastructure.

[1] OARC hosts equipment for researchers who need additional computing resources.
[2] f-pao1 is in Palo Alto, CA; c-mad1 in Madrid, ES; and m-icn in Incheon, South Korea.
[3] f-cdg1 is in Paris, FR, and f-sjc1 in San Jose, CA.

	DITL2007	DITL2008	DITL2009
Duration	roots, 24h	roots, 24h	roots, 24h
IPv4			
# instances*	C: 4/4 F: 36/40 K: 15/17 M: 6/6	A: 1/1 C: 4/4 E: 1/1 F: 35/41 H: 1/1 K: 16/17 L: 2/2 M: 6/6	A: 1/1 C: 6/6 E: 1/1 F: 35/48 H: 1/1 K: 16/17 L: 2/2 M: 6/6
# queries	3.83 B	7.99 B	8.09 B
# clients	2.8 M	5.6 M	5.8 M
IPv6			
# instances*	F: 5/40 K: 1/17	F: 10/41 H: 1/1 K: 1/17 M: 4/6	F: 16/48 H: 1/1 K: 9/17 M: 5/6
# queries	0.2 M	23 M	29 M
# clients	60	9 K	16 K
TCP			
# instances*	C: 4/4 F: 36/40 K: 14/17 M: 5/6	A: 1/1 E: 1/1 F: 35/41 K: 16/17 M: 5/6	A: 1/1 C: 6/6 E: 1/1 F: 35/48 H: 1/1 K: 16/17 M: 5/6
# query	0.7 M	2.07 M	3.04 M
# client	256 K	213 K	163 K
*observed/total			

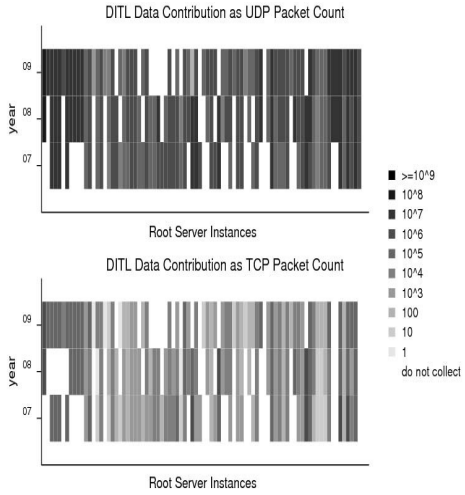


Fig. 1. DITL data coverage for 2007, 2008, 2009. The table summarizes participating root instances, and statistics for the most complete 24-hour collection intervals, including IPv4 UDP, IPv6 UDP, and TCP packets. The plots on the right show data collection gaps for UDP and TCP DNS traffic to the roots for the last three years.

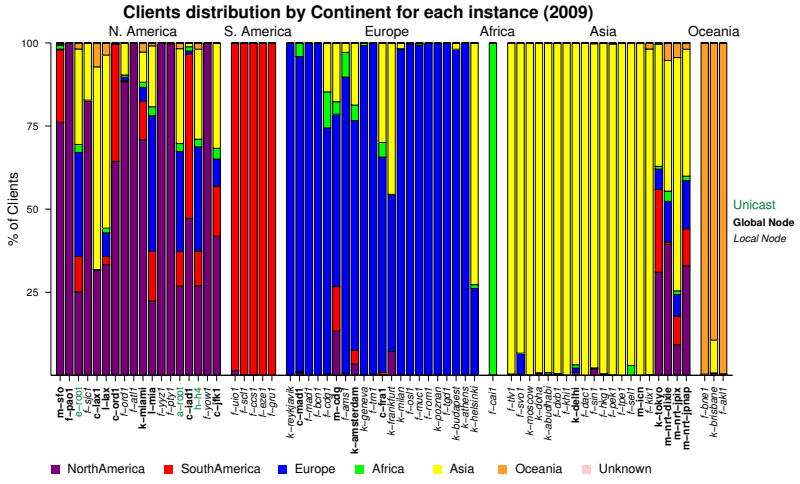
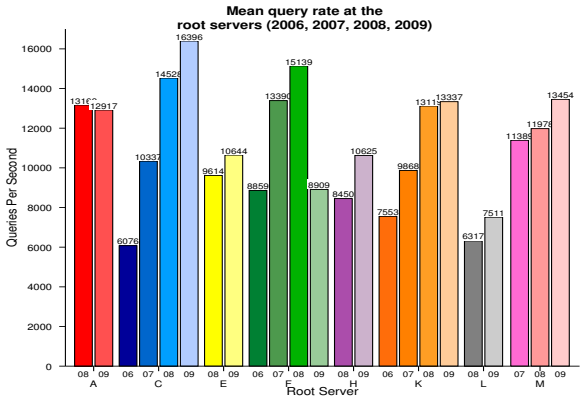


Fig. 2. The geographic distribution of clients querying the root server instances participating in the DITL 2009 (colored according to their continental location). The root server instances are sorted by geographic longitude. Different font styles indicate unicast (green), global anycast (black, bold) and local anycast nodes (black, italic). The figure shows that anycast achieves its goal of localizing traffic, with 42 out of 46 local anycast instances indeed serving primarily clients from the same continent.



Root	Growth 2007-2008	Growth 2008-2009
A		-1.91%
C	40.5%	12.86%
E		10.71%
F	13.06%	-41.15%
H		25.74%
K	32.94%	1.66%
L		18.90%
M	5.17%	12.32%

Fig. 3. Mean query rate over IPv4 at the root servers participating in DITL from 2006 to 2009. Bars represent average query rates on eight root servers over the four years. The table presents the annual growth rate at participating root servers since 2007. The outlying (41%) negative growth rate for F-root is due to a measurement failure at (and thus no data from) a global F-root (F-SFO) node in 2009.

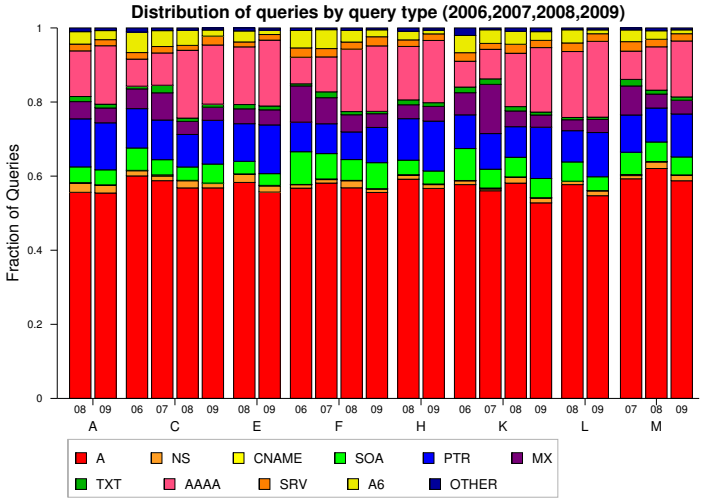


Fig. 4. DITL distribution of IPv4 UDP queries by types from 2007 to 2009. IPv6-related developments caused two notable shifts in 2008: a significant increase in AAAA queries due to the addition of IPv6 glue records to root servers, and a noticeable decrease in A6 queries due to their deprecation.

Figure 4 shows that the most common use of DNS – requesting the IPv4 address for a hostname via *A-type queries* – accounts for about 60% of all queries every year. More interesting is the consistent growth (at 7 out of 8 roots) in AAAA-type queries, which map hostnames to IPv6 addresses, using IPv4 packet transport. IPv6 glue records were added to six root servers in February 2008, prompting a larger jump in 2008 than we saw this year. Many client resolvers, including BIND, will proactively look for IPv6 addresses of NS records, even if they do not have IPv6 configured locally. We further discuss IPv6 in Section 5.

Figure 4 also shows a surprising drop in MX queries from 2007 to 2009, even more surprising since the number of clients sending MX queries increased from .4M to 1.4M over the two data sets. The majority of the moderate to heavy hitter "MX" clients dramatically reduced their per-client MX load on the root system, suggesting that perhaps spammers are getting better at DNS caching.

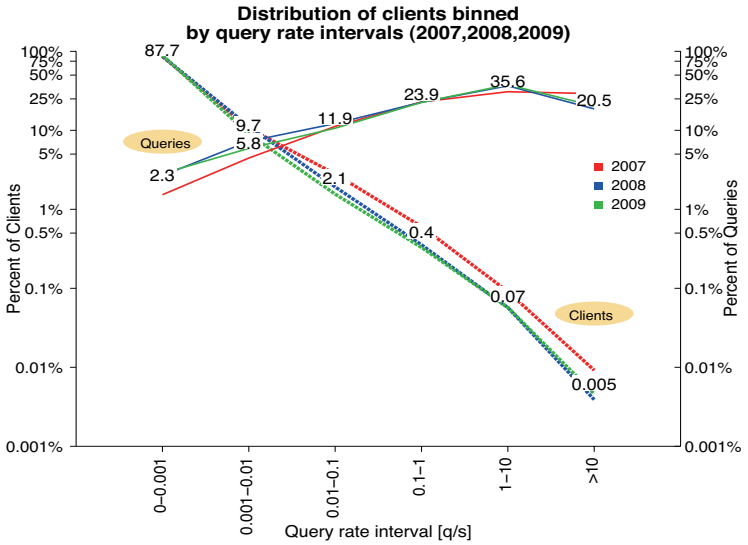


Fig. 5. Distribution of clients and queries as a function of mean IPv4 query rate order of magnitude for last three years of DITL data sets (y-axes log scale), showing the persistence of heavy-hitters, i.e. a few clients (in two rightmost bins) account for more than 50% of observed traffic. The numbers on the lines are the percentages of queries (upward lines) and clients represented by each bin for DITL 2009 (24-hour) data.

Several aspects of client query rates are remarkably consistent across years: the high variation in rate, and the distributions of clients and queries as a function of query rate interval. We first note that nameservers cache responses, including referrals, conserving network resources so that intermediate servers do not need to query the root nameservers for every request. For example, the name server learns that *a.gtld-servers.net* and others are authoritative for the *.com* zone, but alsos learns a *time-to-live* (TTL) for which this information is

considered valid. Typical TTLs for top level domains are on the order of 12 days. In theory, a caching recursive nameserver only needs to query the root nameservers for an unknown top level domain or when a TTL expires. However, many previous studies have shown that the root nameservers receive many more queries than they should [23, 22, 13, 7].

Figure 5 shows the distributions of clients and queries binned by average query rate order of magnitude, ranging from 0.001 q/s (queries per second) to >10 q/s. The decreasing lines show the distribution of clients (unique IP addresses) as a function of their mean query rate (left axis), and the increasing lines show the distribution of total query load produced by clients as a function of their mean query rate (right axis). The two bins with the lowest query rates (under 1 query per 100s) contain 97.4% of the clients, but are only responsible for 8.1% of all queries. In stark contrast, the busiest clients (more than 1 query/sec) are miniscule in number (<0.08%, or 5483 client IPs) but account for 56% of the total query load.

Table 1. The number and fraction of clients, queries, and valid queries in each query rate interval, for a 10% random sample of DITL2009 clients for each root.

Rate interval	Number of clients	Number of queries	Number of valid queries
<0.001	602 K	23 M (2.7%)	8,088 K (47.9%)
0.001-0.01	72 K	49 M (5.7%)	5,446 K (32.3%)
0.01-0.1	14 K	79 M (9.2%)	2,343 K (13.9%)
0.1-1	3 K	165 M (19.3%)	770 K (4.6%)
1-10	565	324 M (37.8%)	206 K (1.2%)
>10	71	216 M (25.2%)	32 K (0.2%)

We next explore the nature of traffic from these hyperbusy clients, which (still) generate mostly DNS pollution in the form of invalid queries. Given the role of caching DNS responses described above, and the far less consistent implementation of caching of negative (NXDOMAIN) results, a high fraction of invalid queries landing at the root is not too surprising – everything else is more consistently cached. Less expected is the extremely high rate of invalid queries, including identical and repeated queries. We believe this behavior is largely due to a combination of firewalls/middleboxes blocking responses and aggresive retransmission implementations at senders behind these firewalls, as described in RFC 4697[12].

Similar to our previous analyses [23, 7], we categorized DNS root pollution into nine groups i.e. (i) unused query class; (ii) A-for-A queries; (iii) invalid TLD; (iv) non-printable characters; (v) queries with '_'; (vi) RFC 1918 PTR [15]; (vii) identical queries; (viii) repeated queries; and (ix) referral-not-cached queries. We classify the remaining queries as legitimate. Since some of the pollution categories require keeping state across the trace, computational limitations prevented us from analyzing pollution for the entire 24-hour traces. Table 1 reflects a set of queries from a random sample of 10% clients for each root in the 2009 dataset.

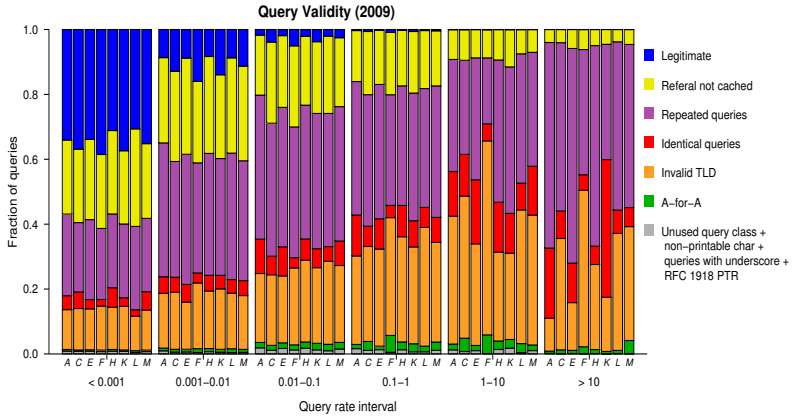


Fig. 6. Query validity as a function of query rate (2009) of the reduced datasets (queries from a random 10% sample of clients)

Figure 6 reflects this sample set of queries and confirms previous years – over 98% of these queries are pollution. The three rightmost groups in Figure 6 and corresponding three bottom rows of Table 1, which include moderately and very busy clients, represent less than 0.54% of the client IPs, but send 82.3% of the queries observed, with few legitimate queries.

A closer look at the pollution class of invalid TLDs (orange bars in Figure 6) reveals that the top 10 most common invalid TLDs represent 10% of the total(!) query load at the root servers, consistently over last four years. The most common invalid TLD is always *local*, followed by (at various rankings within the top 10) generic TLD names such as *belkin, lan, home, invalid, domain, localdomain, wpad, corp* and *localhost*, suggesting that misconfigured home routers contribute significantly to the invalid TLD category of pollution.

Table 2. Pollution and total queries of the busiest DITL2009 clients

Clients	% of clients	#Pollution/#Total	% of queries
Top 4000	0.07%	4,958M/4,964M=99.9%	61.39%
Top 4000-8000	0.07%	760M/ 762M=99.7%	9.42%
Top 8000-32000	0.41%	1,071M/1,080M=99.2%	13.36%
Top 32000	0.55%	6,790M/6,803M=99.8%	84.13%
All clients	100.00%	#Total queries: 8,086M	100.00%

To explore whether we can safely infer that the 98% pollution in our sample also reflects the pollution level in the complete data set, we examine a different sample: the busiest ("heavy hitter") clients in the trace. We found that the 32,000 (0.55%) busiest clients accounted for a lower bound of 84% of the pollution

queries in the whole trace (Table 2). These busy clients sent on average more than 1 query every 10 seconds during the 24-hour interval (the 3 rightmost groups in Figures 5 and 6). We also mapped these busy clients to their origin ASes, and found no single AS was responsible for a disproportionate number of either the busy clients or queries issued by those clients. DNS pollution is truly a pervasive global phenomenon. There is considerable speculation on whether the impending changes to the root will increase the levels and proportion of pollution, and the associated impact on performance and provisioning requirements. Again, the DITL data provide a valuable baseline against which to compare future effects.

4 Security-related attributes of DNS clients

We next explore two client attributes related to DNS security and integrity.

4.1 Source Port Randomness

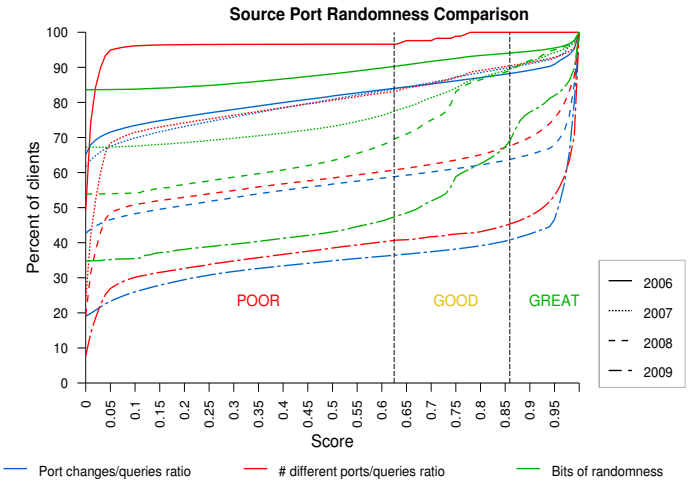


Fig. 7. CDFs of Source Port Randomness scores across four years of DITL data. Scores <0.62 are classified as *Poor*, scores in [0.62, 0.86] as *Good* and scores >0.86 as *Great*. DNS source port randomness has increased significantly in the last 4 years, with the biggest jump between 2008 and 2009, likely in response to Kaminksy's demonstration of the effectiveness of port-guessing to poison DNS caches [17].

The lack of secure authentication of either the DNS mapping or query process has been well-known among researchers for decades, but a discovery last year by

Dan Kaminksy [17] broadened consciousness of these vulnerabilities by demonstrating how easy it was to poison (inject false information info) a DNS cache by guessing port numbers on a given connection[4]. This discovery rattled the networking and operational community, who immediately published and promoted tools and techniques to test and improve the degree of randomization that DNS resolvers apply to DNS source ports. Before Kaminsky's discovery, DITL data indicated that DNS port randomization was typically poor or non-existent [7]. We applied three scores to quantify the evolution of source port randomness from 2006-2009. For each client sending more than 20 queries during the observation interval, we calculated: (i) the number of port number changes/query ratio; (ii) the number of unique ports/query ratio; (iii) bits of randomness as proposed in [21, 22]. We then classified scores <0.62 as *Poor*, in the range $[0.62, 0.86]$ as *Good*, and scores >0.86 as *Great*. Figure 7 shows some good news: scores improved significantly, especially in the year following Kaminsky's (2008) announcement. In 2009, more than 60% of the clients changed their source port numbers between more than 85% of their queries, which was only the case for about 40% of the clients in 2008 and fewer than 20% in 2007.

4.2 DNSSEC capability

Although source-port randomization can mitigate the DNS cache poisoning vulnerability inherent in the protocol, it cannot completely prevent hijacking. The longer-term solution proposed for this vulnerability is the IETF-developed DNS Security extensions (DNSSEC) [3] architecture and associated protocols, in development for over a decade but only recently seeing low levels of deployment [19]. DNSSEC adds five new resource record (RR) types: Delegation signer (DS), DNSSEC Signature (RRSIG), Next-Secure record NSEC and NSEC3), and DNSSEC key request (DNSKEY). DNSSEC also adds two new DNS header flags: Checking Disabled (CD) and Authenticated Data (AD). The protocol extensions support signing zone files and responses to queries with cryptographic keys. Because the architecture assumes a single anchor of trust at the root of the naming hierarchy, pervasive DNSSEC deployment is blocked on cryptographically signing the root zone. Due to the distributed and somewhat convoluted nature of control over the root zone, this development has lagged expectations, but after considerable pressure and growing recognition of the potential cost of DNS vulnerabilities to the global economy, the U.S. government, ICANN, and Verisign are collaborating to get the DNS root signed by 2010. A few countries, including Sweden and Brazil, have signed their own ccTLD's in spite of the root not being signed yet, which has put additional pressure on those responsible for signing the root.

Due to the way DNSSEC works, clients will not normally issue queries for DNSSEC record types; rather, these records are automatically included in responses to normal query types, such as A, PTR, and MX. Rather than count

[4] Source port randomness is an important security feature mitigating the risk of different types of spoofing attacks, such as TCP hijacking or TCP reset attacks [20].

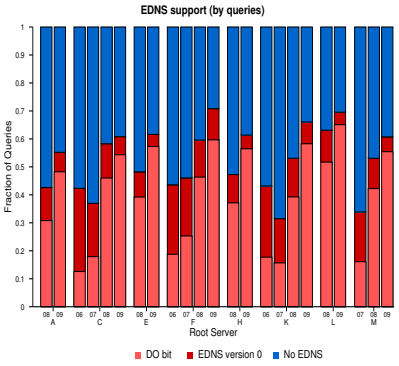


Fig. 8. Growth of EDNS support (needed for DNSSEC) measured by DNS queries, especially between 2007 and 2008. In 2009, over 90% of the EDNS-capable queries are also DO enabled, i.e., advertising DNSSEC capability.

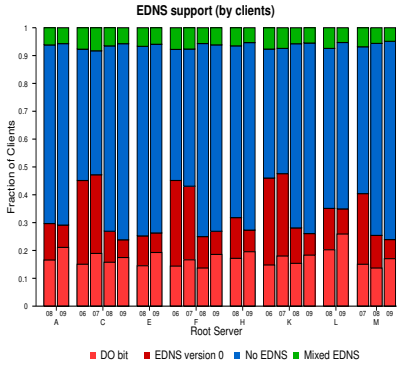


Fig. 9. Decrease in EDNS Support measured by clients. In contrast to the query evolution, the fraction of EDNS enabled clients has dropped since 2007. Worse news for DNSSEC, in 2009 only around 60% of the observed EDNS clients were DO enabled, i.e., DNSSEC-capable.

queries from the set of DNSSEC types, we explore two other indicators of DNSSEC capability across the client population. First we analyse the presence of EDNS support, a DNS extension that allows longer responses, required to implement DNSSEC. We also know that if an EDNS-capable query has its DO bit set, the sending client is DNSSEC-capable. By checking the presence and value of the OPT RR pointer, we classify queries and clients into three groups: (i) no EDNS; (ii) EDNS version 0 (EDNS0) without DO bit set; (iii) and EDNS0 with DO bit. A fourth type of client is *mixed*, i.e. an IP address that sources some, but not all queries with EDNS support. Figure 8 shows clear growth in EDNS support as measured by queries, particularly from 2007 to 2008. Even better news, over 90% of the observed EDNS-capable queries were DO-enabled in 2009. This high level of support for DNSSEC seemed like good news, until we looked at EDNS support in terms of client IP addresses. Figure 9 shows that the fraction of the EDNS-capable clients has actually *decreased* over the last several years, by almost 20%! In 2009, fewer than 30% clients supported EDNS, and of those only around 60% included DO bits indicating actual DNSSEC capability.

We hypothesized that the heavy hitter (hyperbusy) clients had something to do with this disparity, so we grouped clients according to query rate as in Section 3. Figure 10 shows that EDNS support for clients sending few queries dropped significantly after 2007, while busy clients have increased EDNS support. In our 2009 data set, more than half of the EDNS queries were generated by the fewer than 0.1% of clients in the two rightmost categories, sending more than 1 query/sec. (cf. Figure 5). Since we have already determined that these busiest clients generate almost no legitimate DNS queries, we conclude that most of the DNSSEC-capable queries are in pollution categories.

The category of clients with mixed EDNS support represents 7% (or 396K) of the unique sources in the 2009 dataset. We identified two reasons why clients can

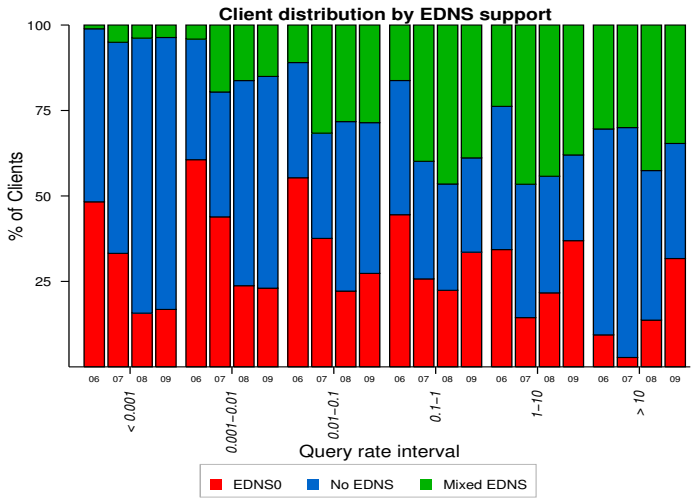


Fig. 10. Plotting EDNS support vs. query rate reveals that EDNS support is increasing for busy clients, who mainly generate pollution, but has declined substantially for low frequency (typical) clients.

show mixed support: (i) several hosts can hide behind the same IP address (e.g. NAT); and (ii) EDNS fallback, i.e. clients fail to receive responses to queries with EDNS support, so they fallback to "vanilla" DNS and retry once more without EDNS support. A test on a sample of 72K (18%) of the mixed EDNS clients showed that EDNS fallback patterns account for 36% of the mixed clients.

EDNS also provides a mechanism to allow clients to advertise UDP buffer sizes larger than the default maximum size of 512 bytes [14]. Traditionally, responses larger than 512 bytes had to be sent using TCP, but EDNS signaling enables transmission of larger responses using UDP, avoiding the potential cost of a query retry using TCP.

Figure 11 shows the UDP buffer size value distribution found in the queries signaling EDNS support. There are only four different values observed: (1) 512 bytes was the default maximum buffer size for DNS responses before the introduction of EDNS in RFC 2671 [18]; 1280 bytes is a value suggested for Ethernet networks to avoid fragmentation; 2048 was the default value for certain versions of BIND and derived products; and 4096 bytes is the maximum value permitted by most implementations.

Figure 11 reveals a healthy increase in the use of the largest buffer size of 4096 bytes (from around 50% in 2006 to over 90% in 2009), which happened at the expense of queries with a 2048-byte buffer size. The fraction of queries using a 512-byte buffer size is generally below 5%, although it sometimes varies over years, with no consistent pattern across roots. One of the deployment concerns surrounding DNSSEC is that older traffic filtering appliances, firewalls, and other

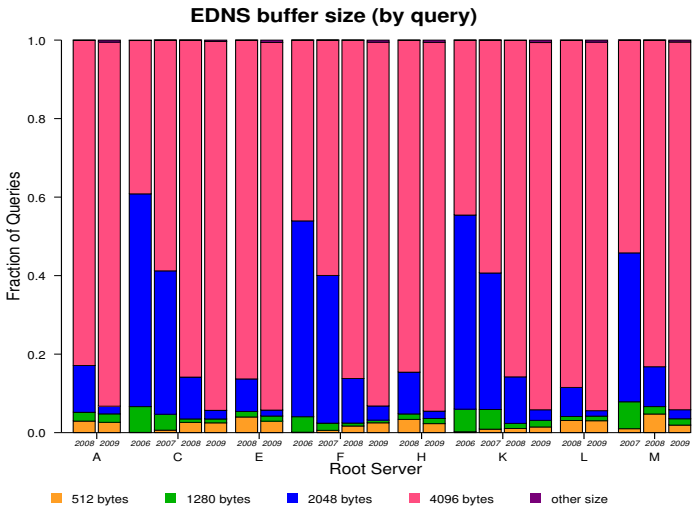


Fig. 11. Another capability provided by EDNS is signaling of UDP buffer sizes. For the queries with EDNS support, we analyze the buffer size announced. An increase from 50% to 90% in the largest size can be observed from 2006 to 2009.

middleboxes may drop DNS packets larger than 512 bytes, forcing operators to manually set the EDNS buffer size to 512 to overcome this limitation. These middleboxes are harmful to the deployment of DNSSEC, since small buffer sizes combined with the signaling of DNSSEC support (by setting the DO bit on) could increase the amount of TCP traffic due to retries.

5 A first look at DNS IPv6 data

Proposed as a solution for IPv4 address exhaustion, IPv6 supports a vastly larger number of endpoint addresses than IPv4, although like DNSSEC its deployment has languished. As of November 2009, eight of the thirteen root servers have been assigned IPv6 addresses [1]. The DITL 2009 datasets are the first with significant (but still pretty inconsistent) IPv6 data collection, from four root servers. Table 3 shows IPv6 statistics for the one instance of K-root (in Amsterdam) that captured IPv6 data, without huge data gaps in the collection, for the last three years. Both the IPv6 query count and unique client count are much lower than for IPv4, although growth in both IPv6 queries and clients is evident. Geolocation of DITL 2009 clients reveals that at least 57.9% of the IPv6 clients querying this global root instance are from Europe [16], not surprising since this instance is in Europe, where IPv6 has had significant institutional support. The proportion of legitimate IPv6 queries (vs. pollution) is 60%, far higher than for IPv4, likely related to its extremely low deployment [4, 11].

Table 3. IPv4 vs. IPv6 traffic on the K-AMS-IX root instance over three DITL years

K-AMS-TX, k-root	2007		2008		2009	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
Query Count	248 M	39 K	170 M	8.21 M	277.56 M	9.96 M
Unique Clients	392 K	48	340 K	6.17 K	711 K	9 K

6 Lessons learned

The Domain Name System (DNS) provides critical infrastructure services necessary for proper operation of the Internet. Despite the essential nature of the DNS, long-term research and analysis in support of its performance, stability, and security is extremely sparse. Indeed, the biggest concern with the imminent changes to the DNS root zone (DNSSEC, new TLDs, and IPv6) is the lack of data with which to evaluate our preparedness, performance, or problems before and throughout the transitions. The DITL project is now four years old, with more participants and types of data each year across many strategic links around the globe. In this paper we focused on a limited slice – the most detailed characterization of traffic to as many DNS root servers possible, seeking macroscopic insights to illuminate the impending architectural changes to the root zone. We validated previous results on the extraordinary high levels of pollution at the root nameservers, which continues to constitute the vast majority of observed queries to the roots. We presented new results on security-related attributes of the client population: an increase in the prevalence of DNS source port randomization, and a surprising decreasing trend in the fraction of DNSSEC-capable clients, which serve as a motivating if disquieting baseline for the impending transition to DNSSEC.

From a larger perspective, we have gained insights and experience from these global trace collection experiments, which inspire recommended improvements to future measurements that will help optimize the quality and integrity of data in support of answering critical questions in the evolving Internet landscape. We categorize our lessons into three categories: data collection, data management, and data analysis.

Lessons in Data Collection Data collection is hard. Radically distributed Internet data collection across every variety of administrative domain, time zone, and legislative framework around the globe is in "pray that this works" territory. Even though this was our fourth year, we continued to fight clock skew, significant periods of data loss, incorrect command line options, dysfunctional network taps, and other technical issues. Many of these problems we cannot find until we analyze the data.

We rely heavily on pcap for packet capture and have largely assumed that it does not drop a significant number of packets during collection. We do not know for certain if, or how many, packets are lost due to overfull buffers or other external reasons. Many of our contributors use network taps or SPAN ports, so it is possible that the server receives packets that our collector

does not. Next year we are considering encoding the *pcap_stats*() output as special "metadata" packets at the end of each file.

For future experiments, we also hope to pursue additional active measurements to improve data integrity and support deeper exploration of questions, including sending timestamp probes to root server instances during collection interval to test for clock skew, fingerprinting heavy hitter clients for additional information, and probing to assess extent of DNSSEC support and IPv6 deployment of root server clients. We recommend community workshops to help formulate questions to guide others in conducting potentially broader "Day-in-the-Life" global trace collection experiments [6].

Lessons in Data Management As DITL grows in number and type of participants, it also grows in its diversity of data "formatting". Before any analysis can begin, we spend months fixing and normalizing the large data set. This curation includes: converting from one type of compression (lzop) to another (gzip), accounting for skewed clocks, filling in gaps of missing data from other capture sources[5], ensuring packet timestamps are strictly increasing, ensuring pcap files fall on consistent boundaries and are of a manageable size, removing packets from unwanted sources[6], separating data from two sources that are mixed together[7], removing duplicate data[8], stripping VLAN tags, giving the pcap files a consistent data link type, removing bogus entries from truncated or corrupt pcap files. Next, we merge and split pcap files again to facilitate subsequent analysis.

The establishment of DNS-OARC also broke new (although not yet completely arable) ground for disclosure control models for privacy-protective data sharing. These contributions have already transformed the state of DNS research and data-sharing, and if sustained and extended, they promise to dramatically improve the quality of the lens with which we view the Internet as a whole. But methodologies for curating, indexing, and promoting use of data could always use additional evaluation and improvement. Dealing with extremely large and privacy-sensitive data sets remotely is always a technical as well as policy challenge.

Lessons in Data Analysis We need to increase the automatic processing of basic statistics (query rate and type, topological coverage, geographic characteristics) to facilitate overview of traces across years. We also need to extend our tools to further analyze IPv6, DNSSEC, and non-root server traces to promote understanding of and preparation for the evolution of the DNS.

[5] In 2009, for example, one contributor used *dnscap*, but their scripts stopped working. They also captured data using *WDCAP* and were able to fill in some gaps, but the WDCAP data files were not boundary-aligned with the missing pcap files.

[6] Another contributor included packets from their nearby non-root nameservers.

[7] In 2008, we received data from A-root and Old-J-Root as a single stream.

[8] In 2007-09, at least one contributor mistakenly started two instances of the collection script.

References

1. List of root servers. http://www.root-servers.org/ (accessed 2009.11.20).
2. NetAcuity. http://www.digital-element.net (accessed 2009.11.20).
3. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, 2005.
4. CAIDA. Visualizing IPv6 AS-level Internet Topology, 2008. http://www.caida.org/research/topology/as_core_network/ipv6.xml (2009.11.20).
5. CAIDA and DNS-OARC. A Report on DITL data gathering Jan 9-10th 2007. http://www.caida.org/projects/ditl/summary-2007-01/ (accessed 2009.11.20).
6. CAIDA/WIDE. What researchers would like to learn from the ditl project, 2008. http://www.caida.org/projects/ditl/questions/ (accessed 2009.11.20).
7. S. Castro, D. Wessels, M. Fomenkov, and k. claffy. A Day at the Root of the Internet. In *ACM SIGCOMM Computer Communications Review (CCR)*, 2008.
8. N. R. Council. *Looking over the Fence: A Neighbor's View of Networking Research*. National Academies Press, 2001.
9. DNS-OARC. DNS-DITL 2009 participants. https://www.dns-oarc.net/oarc/data/ditl/2009 (2009.11.20).
10. S. Gibbard. Observations on Anycast Topology and Performance, 2007. http://www.pch.net/resources/papers/anycast-performance/anycast-performance-v10.pdf (2009.11.20).
11. E. Karpilovsky, A. Gerber, D. Pei, J. Rexford, and A. Shaikh. Quantifying the Extent of IPv6 Deployment. In *Passive and Active Measurement Conference (PAM) '09*, Seoul, Korea, 2009.
12. M. Larson and P. Barber. Observed DNS Resolution Misbehavior. RFC 4697, 2006.
13. Z. Liu, B. Huffaker, N. Brownlee, and kimberly claffy. Two Days in the Life of the DNS Anycast Root Servers. In *Passive and Active Measurement Conference (PAM) '07*, pages 125–134, Louvain-la-Neuve, Belgium, 2007.
14. P. Mockapetris. Domain names - implementation and specification. RFC 1035 (Standard), 1987.
15. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, 1996.
16. Team Cymru. Ip to asn mapping. http://www.team-cymru.org/Services/ip-to-asn.html (accessed 2009.11.20).
17. US-CERT. Vulnerability note vu#800113: Multiple dns implementations vulnerable to cache poisonings. http://www.kb.cert.org/vuls/id/800113 (2009.11.20.
18. P. Vixie. Extension Mechanisms for DNS (EDNS0). RFC 2671, 1999.
19. P. Vixie. Reasons for deploying DNSSEC, 2008. http://www.dnssec.net/why-deploy-dnssec (2009.11.20).
20. P. Watson. Slipping in the Window: TCP Reset attacks, 2004. http://osvdb.org/ref/04/04030-SlippingInTheWindow_v1.0.doc (2009.11.20).
21. D. Wessels. DNS port randomness test. https://www.dns-oarc.net/oarc/services/dnsentropy (2009.11.20).
22. D. Wessels. Is your caching resolver polluting the internet? *ACM SIGCOMM Workshop on Network Troubleshooting (Netts) '04*, 2004.
23. D. Wessels and M. Fomenkov. Wow, that's a lot of packets. In *Passive and Active Measurement Workshop (PAM) '02*, Fort Collins, USA, 2002.