DNS Observatory: The Big Picture of the DNS

Pawel Foremski
Farsight Security, Inc. / IITiS PAN
pjf@fsi.io

Oliver Gasser
Technical University of Munich
gasser@net.in.tum.de

Giovane C. M. Moura
SIDN Labs / TU Delft
giovane.moura@sidn.nl

ABSTRACT

The Domain Name System (DNS) is thought of as having the simple-sounding task of resolving domains into IP addresses. With its stub resolvers, different layers of recursive resolvers, authoritative nameservers, a multitude of query types, and DNSSEC, the DNS ecosystem is actually quite complex.

In this paper, we introduce DNS Observatory: a new stream analytics platform that provides a bird's-eye view on the DNS. As the data source, we leverage a large stream of passive DNS observations produced by hundreds of globally distributed probes, acquiring a peak of 200 k DNS queries per second between recursive resolvers and authoritative nameservers. For each observed DNS transaction, we extract traffic features, aggregate them, and track the top-k DNS objects, e.g., the top authoritative nameserver IP addresses or the top domains.

We analyze 1.6 trillion DNS transactions over a four month period. This allows us to characterize DNS deployments and traffic patterns, evaluate its associated infrastructure and performance, as well as gain insight into the modern additions to the DNS and related Internet protocols. We find an alarming concentration of DNS traffic: roughly half of the observed traffic is handled by only 1 k authoritative nameservers and by 10 AS operators. By evaluating the median delay of DNS queries, we find that the top 10 k nameservers have indeed a shorter response time than less popular nameservers, which is correlated with less router hops.

We also study how DNS TTL adjustments can impact query volumes, anticipate upcoming changes to DNS infrastructure, and how negative caching TTLs affect the Happy Eyeballs algorithm. We find some popular domains with a a share of up to 90 % of empty DNS responses due to short negative caching TTLs. We propose actionable measures to improve uncovered DNS shortcomings.

CCS CONCEPTS

• **Networks** → **Naming and addressing**; **Network measurement**; • **Information systems** → *Data stream mining*.

ACM Reference Format:
Pawel Foremski, Oliver Gasser, and Giovane C. M. Moura. 2019. DNS Observatory: The Big Picture of the DNS. In *Internet Measurement Conference (IMC '19), October 21–23, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3355369.3355566

IMC '19, October 21–23, 2019, Amsterdam, Netherlands
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6948-0/19/10…$15.00
https://doi.org/10.1145/3355369.3355566

1 INTRODUCTION

Although the DNS dates back to 1983, when Mockapetris published its original specification in RFCs 882 and 883 [45, 46], it still remains one of the key protocols of the Internet. Since then, various authors have published a staggering number of 3200 pages of RFC documents (counting Internet Standard, Proposed Standard, and Informational documents), which demonstrates how deceptive it is to think that DNS is simple, well understood, or already studied enough [31–33, 67]. In this context, we see Internet engineers struggling for a faster, better connected, and more secure Web—through the adoption of IPv6, HTTP/2, QUIC, and TLS 1.3—all of which are directly affected by the DNS. We believe that big-scale measurements of the DNS *in the wild* are essential to understand and revise it, so that the DNS stays on par with the improvements made to other key protocols of the Internet.

Modern, large-scale authoritative DNS servers employ high levels of complexity. First, they employ IP anycast [1], meaning that the same prefix can be announced from multiple locations around the globe. Secondly, the contents of a DNS response may depend on where the user is located: authoritative servers can be configured to give different DNS answers based on geo-location, latency, and content filtering policy [6]. As such, any researchers attempting to evaluate DNS will—depending on their *vantage point*—have only a *partial* view of a DNS zone. Besides, whenever they employ active measurements on DNS zones, they have to actively query for pre-obtained domain lists, which skews the results even more [60].

This paper introduces DNS Observatory, a novel stream analytics platform that mitigates the issue of vantage point location by collecting data from hundreds of DNS resolvers distributed around the world, and which stores only aggregate information extracted from the traffic between resolvers and authoritative nameservers.

We analyze the data collected in DNS Observatory from January until April 2019, totaling 1.6 trillion DNS transactions, and report on our findings. We present the Big Picture of the DNS, which helps us to better understand DNS traffic distributions, global DNS performance, impact and dynamics of the TTLs, and the possible consequences of the Happy Eyeballs algorithm on IPv4-only sites, due to negative caching misconfigurations. We believe our work can help in making informed improvements to the DNS, and to bolster DNS research in general.

Our main contributions include:

DNS Observatory This work presents DNS Observatory (DO), which is built on hundreds of globally distributed resolver vantage points (VPs). It aggregates up to 200 k DNS queries per second in a stream of top-k DNS objects, which can be used for various analyses. We elaborate on the design and architecture of DO in section 2.

Big Picture We analyze the big picture of DNS in terms of traffic distributions, query types, response delays, Autonomous

Systems, and QNAME minimization deployment in section 3. We find that 50% of observed DNS traffic is likely handled by just top 1,000 nameservers, and by IP prefixes managed by just 10 organizations.

Dissecting TTLs We perform an in-depth analysis of TTLs, specifically their correlation with traffic and infrastructure changes in section 4.

Happy Eyeballs and Negative Caching We evaluate the effect of low negative caching TTLs and the Happy Eyeballs algorithm used by IPv6 enabled clients in section 5. We find some domains with more than 90 % of all responses being empty due to low negative caching TTLs. Finally, we propose actionable steps to ameliorate the current state.

We begin the paper by introducing terminology, describing raw DNS traffic data sources, and elaborating on methodology in Section 2. Then, in Sections 3, 4, and 5 we present our findings on the DNS. In Section 6, we refer the reader to related works and datasets. We conclude in Section 7, inviting academic researchers to access the data collected in DNS Observatory.

2 DNS OBSERVATORY

In this section, we present our methodology, i.e., our data processing pipeline, which involves: obtaining raw data, preprocessing, tracking objects, measuring DNS traffic, and producing time series data for various time aggregations. The overall design of DNS Observatory is presented in Figure 1.

Before we describe our system in detail, we make a note on the terminology used in this paper. We use the terms "resolver" and "nameserver" to refer to a particular IP address used by a recursive DNS resolver and authoritative DNS name server, respectively. The terms "Top-Level Domain" (TLD) and "Second-Level Domain" (SLD) refer to, respectively, the last 1 and the last 2 labels in Fully-Qualified Domain Names (FQDNs). The term "effective TLDs" (eTLDs) refer to the ICANN domains listed in the Public Suffix List [49] (e.g., `.co.uk`), and "effective SLD" (eSLD) is simply a label directly under an eTLD (e.g., `bbc.co.uk`).

2.1 Preprocessing raw data

Our data comes from a large stream of *passive* observations of DNS traffic between recursive resolvers and authoritative nameservers, i.e., essentially we analyze the DNS cache-miss query-response transactions *above* DNS resolvers. The stream comes from the Security Information Exchange (SIE), an open platform operated by Farsight Security, Inc.. SIE resembles an Internet Exchange Point (IXP) and allows the cooperating parties to exchange Internet security information in real-time [22].

DNS Observatory ingests the main passive DNS stream available on the platform, which peaks at 200 k DNS transactions per second and comes from hundreds of resolvers. The data is contributed by many parties spread around the world, including: North and Central America, Western Europe and UK, Eastern Europe and Russia, Middle East, Southeast Asia, and Australia. The contributors include large ISPs, recursive DNS providers, hosting farms, social media platforms, Internet security companies, universities, financial institutions, etc. The data is generated by open source software—usually deployed directly on the resolver machine—that reconstructs the DNS transactions by capturing raw IP packets from network interfaces [20].

In more detail, we analyze UDP/53 transactions with either both the query and the response packets, or the query alone (in case of no response). Support for TCP/53 is planned as future work, but note that it constitutes only <3% of Internet traffic [36, 63]. Each transaction includes raw packets, starting at the IP header, and detailed timestamps. Transactions are serialized using Protocol Buffers [29] and submitted to the platform. From there, we read the stream, deserialize the data, parse IP headers and DNS payloads, and summarize each transaction with a line of text.

We retain only the relevant pieces of information, e.g., resolver and nameserver IP address, response delay, DNS header contents, queried name, and select DNS record data. Our goal is to make the data easier to process in the next steps, given the data volume.

2.2 Tracking Top-k objects

The basic tool we use for DNS Observatory is the Space-Saving algorithm (SS) [44], which allows us to keep track of the most frequently queried nameservers—or, in more general, to continuously track the Top-k DNS *objects* in our data stream—while keeping memory usage under control.

A DNS object is any entity within the DNS, identified with a textual *key*: the value of any transaction detail, or a combination thereof. For example—although the primary objects we consider are the nameserver IP addresses—we may study the most popular FQDNs, returned IP addresses, or combinations of the FQDN and its IP address. When desirable, we may also filter the input traffic, e.g., consider only the transactions involving root nameservers, or the responses with the Authoritative Answer (AA) flag set.

Note that usually the number of DNS objects is too large to keep track of all of them, e.g., all FQDNs seen in DNS. However, their distributions are often heavy-tailed, i.e., a relatively small number of the most frequent objects cover a large fraction of all observed DNS transactions. Thus, the usage of the SS algorithm allows us to obtain a big picture of the DNS.

When a new transaction is observed, we extract its key (e.g., the nameserver IP address) and check if the corresponding object exists in the SS cache. If yes, we update its frequency estimate—an exponentially decaying moving average that tracks the rate of transactions per second for this object. If no, we evict the least frequent object, and insert the new object instead, but keeping (and updating) the frequency estimate of the evicted entry. In the latter case, we optionally consult a Bloom Filter [7] before doing the eviction, in order to skip incidental observations of rare keys.

Note that although at this point we already know the estimated rates of traffic for each object, e.g., hits per second for a nameserver IP address, we do not use them after this step. We only maintain a list of the currently most popular objects in the input data stream. Each live object in the SS cache has a separate state used for traffic statistics, which we update in the next step.

2.3 Measuring traffic features

Each transaction ends up either being aggregated in statistics of a particular DNS object from the SS cache, or being dropped in case the corresponding object is not in the cache.

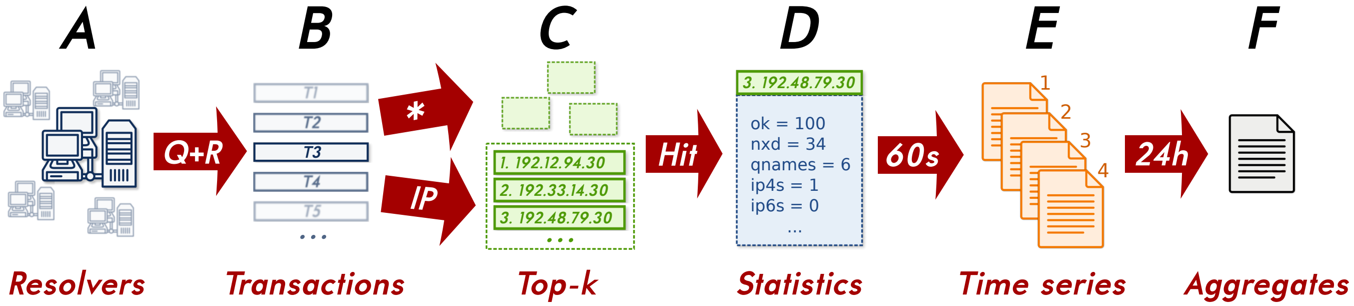

Figure 1: DNS Observatory data processing pipeline: *A*) recursive resolvers submitting cache-miss traffic (subsection 2.1); *B*) summarizing DNS query-response transactions (subsection 2.1); *C*) tracking Top-*k* objects for given key definition, e.g., nameserver IP address (subsection 2.2); *D*) collecting statistics in time windows of 60 seconds (subsection 2.3); *E*) writing time series to disk (subsection 2.4); *F*) aggregating in time, e.g., producing daily files after 24 hours (subsection 2.4).

We characterize each object in the cache using *traffic features*, enumerated and briefly documented below:

- **srvips**: number of nameserver IP addresses;
- **srcips**: number of recursive resolver IP addresses;
- **sources**: number of SIE contributors that saw this object;
- **hits**: total number of transactions seen so far;
- **unans**: number of unanswered queries;
- **ok, nxd, rfs, fail**: responses with the RCODE of respectively: NoError, NXDOMAIN, Refused, and ServFail;
- **ok_ans, ok_ns, ok_add**: NoError responses: non-empty ANSWER section, non-zero NS records in AUTHORITY, non-empty ADDITIONAL section (skipping EDNS0 OPT);
- **ok_nil**: neither **ok_ans** nor **ok_ns** satisfied (NoData);
- **ok6, ok6nil**: AAAA queries: all NoError vs. NoData;
- **ok_sec**: DNSSEC-signed responses: EDNS0 DO flag set, **ok_ans** or **ok_ns** satisfied, sections have RRSIG records;
- **qnamesa, qnames**: number of distinct QNAMEs in all queries vs. those that resulted in a NoError response;
- **tlds, eslds**: number of Top-Level and effective Second-Level Domains in NoError responses;
- **qtypes**: number of QTYPEs in all queries;
- **qdots, lvl, nslvl**: number of labels in all QNAMEs, records in ANSWER, NS records in AUTHORITY, respectively;
- **ip4s, ip6s**: number of distinct IPv4/IPv6 addresses in NoError responses to A/AAAA or ANY queries;
- **ttl, nsttl**: the top-3 TTL values (and distributions) for records in ANSWER and nameservers in AUTHORITY;
- **resp_delays**: quartiles of server response delays;
- **network_hops**: quartiles of inferred number of network hops (routers) between resolvers and nameservers [39];
- **resp_size**: quartiles of the response packet sizes.

The underlying data structure for each feature is either a simple counter (e.g., **hits**), an average (e.g., **qdots**), a histogram (e.g., **resp_delays**), or a cardinality estimate (e.g., **ip4s**). For estimating the number of elements in possibly large sets of values (e.g., **qnamesa**) we use the HyperLogLog algorithm, as improved in [30].

2.4 Producing time series

Every 60 seconds, we dump all data to disk and reset all statistics, but without affecting the SS cache, i.e., we keep the list of the most popular objects, but we clear their internal state used for traffic features. This way we produce time series data that characterize a select aspect of the DNS minute by minute, e.g., a time series of queries per minute for the world's most popular nameservers.

Because the popularity of objects may change at arbitrary points in time (not synchronized with our 1-minute time ticks), we skip the data from objects recently inserted in the SS cache. That is, if we included an object in the data dump, this means it survived the SS cache eviction for 60 seconds.

A separate process aggregates minutely files into new, decaminutely files that represent 10-minute time windows. These in turn get aggregated into hourly files, then into daily files, then into monthly files, and finally into yearly files. In order to keep disk usage under control, each of these time granularities have a data retention policy, i.e., after some time we delete old files for short time windows, and keep only the longer aggregates.

In general, we aggregate time series of a particular feature using the arithmetic mean. The value of a counter feature for a particular object in a decaminutely file is the average rate per minute, estimated using 10 data points (each 1 minute apart); for the hourly file, we use 6 data points (each 10 minutes apart), etc. If the object is missing in some of the files being aggregated, we use a value of 0 for counters. For features that are not counters (e.g., cardinality estimates), we just skip the missing data point.

The data is stored on disk in the TSV file format, where the file name encodes both the time granularity, and the moment of time when we started collecting the data. The first TSV row contains column names, and the last row contains data collection statistics, which include the total number of DNS transactions seen before and after filtering. TSV files may easily be imported into many data analysis tools, from ordinary spreadsheet software to time series databases. For this paper, we mainly used Python and the JupyterLab environment.

2.5 Ethical considerations

The intent of DNS Observatory is to track only the big picture of the DNS—e.g., its performance, robustness, and security—and never to track Internet users—i.e., individuals and/or groups of people. We store only highly aggregated data that does not contain Personally Identifiable Information (PII), and as such minimize the risk of violating users' privacy.

There are three layers of user privacy protection involved in our research. First, the raw DNS data is captured *above* recursive resolvers, which means we only see the stream of DNS queries aggregated for all users of a particular resolver—without the user IP address—and only for FQDNs *not* in the resolver cache, which again is shared by all of the resolver clients. Moreover, we do not know the exact locations of resolvers beyond their IP addresses, and each resolver can in general be used by people located anywhere.

Second, early in our data pipeline we drop DNS transaction details except for those that will end up aggregated in traffic statistics described in subsection 2.3. This means we drop possibly sensitive EDNS0 data, including DNS cookies [17] and client subnet information [12]. The detailed timestamps of the query and response packets are used only to compute the nameserver response delay and are subsequently dropped.

Third, we aggregate traffic in Top-*k* lists for various DNS objects, as described in subsection 2.2. The *k* parameter is finite and relatively small compared with, e.g., the cardinality of the DNS FQDN space. Given the global reach of our raw data collection system, this means a particular object must be popular enough—compared with the rest of the world—in order to survive the SS cache eviction for 60 seconds. In an unlikely event it does survive for short periods of time, the final time aggregation step described in subsection 2.4 will drop it when producing, e.g., the hourly or daily files.

To conclude, we believe that these three layers protect user privacy. Besides, as we do not perform active measurements, we do not induce harm on individuals [4, 15, 54].

3 THE BIG PICTURE

In this section we present results from our evaluations based on data from DNS Observatory. Since our data comes from passive observations of real DNS traffic—recorded between hundreds of recursive resolvers spread around the world and over 1 M authoritative nameservers (subsection 3.7)—we report on the Big Picture of the DNS, which is not visible through active measurements, data collected at TLD level, or from a single ISP or recursive resolver operator.

3.1 Collected datasets

In general, we use datasets collected from January 1st 2019 until April 30th 2019, i.e., the first 120 days of 2019. In this time period, we processed over 1.6 *trillion* DNS transactions, i.e., over 13 *billion* per day. On average, in a 1-minute time window, we see over 1.5 M existing, and 1.1 M non-existing, unique FQDNs.

The measurement process runs without any interruptions, but the capabilities of our system improved with time, which allowed us to collect new aggregations and features starting at a later time. We collected the following datasets:

- **srvip**: Top-100K authoritative nameservers, i.e., transactions aggregated using the nameserver IPv4/IPv6 address;
- **etld**: Top-10K effective TLDs (note that we include NXDOMAIN traffic), i.e., transactions aggregated using the last 1 or more labels in QNAME (since Jan. 8, 2019);
- **esld**: Top-100K effective SLDs, i.e., transactions aggregated using the last 2 or more labels in QNAME (since Feb. 19, 2019);
- **qname**: Top-100K FQDNs, i.e., transactions aggregated using the full QNAME (since Feb. 19, 2019);
- **qtype**: All QTYPE aggregations (since Feb. 15, 2019);
- **rcode**: All RCODE aggregations (since Apr. 10, 2019);
- **aafqdn**: Top-20K FQDNs in authoritative answers (cf. subsection 4.2), i.e., QNAME in transactions where the response has the AA flag set (since Apr. 15, 2019);
- **srcsrv**: Top-30K pairs of resolvers and nameservers (cf. subsection 2.1), i.e., transactions aggregated using the combined IP addresses as key (Apr. 10, 2019 until May 9, 2019);

3.2 Traffic distributions

In Figure 2, we analyze traffic distributions for various Top-*k* aggregations. First, in (a), we consider the 100K most popular nameservers, ranked by their traffic volumes. The data aggregation step described in subsection 2.2 allowed us to capture in this top list 94.9% of all DNS transactions seen in our raw data source. That is, although we skip many unpopular nameservers (see subsection 3.7), we know that they handle only 5.1% of the observed DNS traffic.

In total, all NoError responses account for 68.1% of the transactions captured in the top list, but in the plot we distinguish the NoData (4.7%) and the opposite "NoError + Data" case, i.e., when a successful response either had the answer, or delegated to another nameserver (64.4%). On the other hand, all NXDOMAIN responses account for 20.7% of the top list traffic. For brevity, we skip other RCODEs and unanswered queries, 11.2% in total. Note that we plot an independent CDF curve that ends at 1.0 for each case, so the curves are not to scale with respect to each other.

We find evidence that the majority of observed DNS traffic is likely handled by only ≈1,000 authoritative nameserver IP addresses (i.e., IPv4 and IPv6 addresses). This suggests that considering raw DNS transaction volumes, a big chunk of the DNS is *not* well distributed in the IP address space, and instead relies on shared infrastructure—or at least, on shared addressing—as already shown in [2] from another perspective.

Moreover, the surprising starting point of the NXDOMAIN traffic above 20% is caused by a large botnet, likely "Mylobot" [50]. The botnet's Domain Generation Algorithm (DGA) produced millions of FQDNs under thousands of non-existing SLDs within the .com TLD, which caused spikes of NXDOMAIN traffic towards the gTLD nameservers. This, however, demonstrates how more popular nameservers—usually higher in the DNS hierarchy—are more likely to receive queries for non-existing names, and are thus the DNS's "first line of defence" against artificially generated and otherwise erroneous FQDNs.

In Figure 2 (b), we analyze the list of Top-100K FQDNs, reflecting 23.2% of all DNS transactions seen (the top 10K FQDNs correspond to 18.6% of the observed traffic). Comparing with (a), the lower share simply means that there are much more FQDNs than nameserver IPs in the DNS, and that many FQDNs are ephemeral, i.e., used only once [10]. Thus, we see a heavy-tailed distribution on the plot.

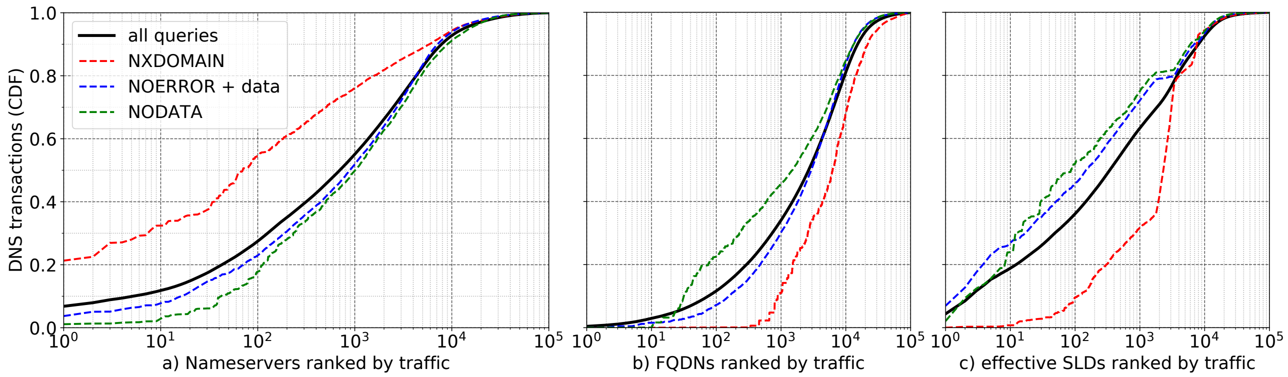

Figure 2: Traffic distributions for various Top-100K DNS objects, ranked by traffic. Note that the x-axis is log-scaled for improved readability.

About 10% of queries captured by the list result in a NoData response, linked to AAAA queries and the Happy Eyeballs algorithm, which we analyze in detail in section 5. The NXDOMAIN traffic (1.5%) is heavily shifted towards less popular FQDNs, which shows the Internet's most popular non-existing—yet queried—names are still well behind the top *existing* FQDNs. The ordinary "NoError + Data" responses correspond to 70.2% of aggregated transactions.

Finally, in Figure 2 (c), we analyze 68.5% of observed DNS transactions, aggregated in a list of Top-100K effective SLDs. The distribution shows a high accumulation of queries towards the Top-100 domains, foremost belonging to the biggest CDNs, cloud providers, social media and e-commerce sites, etc. In addition to the sites already known from various web popularity top lists—e.g., [42]—we found popular domains used by anti-virus software, and by the reverse DNS, both of which are not normally queried when browsing the Web.

NXDOMAIN responses accounted for 18.9% of transactions captured in the list. The shape of CDF curves for ranks 2-4K is due to the botnet already described in (a) above—this time, however, the result is spread on more entries in our top list.

In summary, our results presented in Figure 2 show that a big part of the DNS relies on a relatively small number of authoritative nameserver IPs and domains, which confirms findings by other researchers, e.g., [2, 5].

3.3 Autonomous Systems

In order to evaluate how DNS traffic is distributed on the Autonomous Systems comprising the Internet, we associate each IP address in our Top-100K nameserver list with its corresponding AS number, using the data collected by the University of Oregon's Route Views project [64]. Next, for each ASN, we lookup its name using the AS Names dataset [35]. Finally, we extract the organization name from each AS Name string, and aggregate nameservers in groups based on the result.

We present the top 10 names, ranked by the total volume of DNS transactions in Table 1. The basic observation we make is that the IP prefixes managed by just 10 organizations receive more than half of the world's DNS queries.

	Name	ASes	global	servers	delay	hops
1	AMAZON	3	16%	**5,026**	**60.9**	12.0
2	VERISIGN	7	10%	62	53.5	9.6
3	CLOUDFLARE	2	6.6%	**995**	**26.5**	**6.6**
4	AKAMAI	6	6.4%	**6,844**	**14.9**	**7.3**
5	MICROSOFT	5	2.7%	475	**74.8**	**13.5**
6	PCH	2	2.4%	178	29.9	7.2
7	ULTRADNS	1	2.3%	925	24.6	8.2
8	GOOGLE	1	2.1%	243	**89.9**	**13.3**
9	DYNDNS	1	1.8%	598	56.0	10.5
10	GODADDY	2	1.2%	372	63.0	11

Table 1: Top 10 AS names, by volume of DNS transactions: *Name*) organization name, extracted from WHOIS data; *ASes*) number of matching ASes; *global*) share in observed DNS transactions; *servers*) nameserver IP count; *delay, hops*) average response delay [ms] and hop count. Highlighted values are analyzed in subsection 3.3.

Note that the fact that a particular organization announces a particular IP prefix does not automatically mean it also runs all of the nameservers within that prefix. Most importantly, this is the case for AMAZON, MICROSOFT, and GOOGLE—all of which provide VPS cloud services besides dedicated DNS hosting services—where we see relatively high response delays, correlated with higher hop counts.

Surprisingly, AMAZON leads the table with 16% share in DNS transactions measured in DNS Observatory. This is likely due to >5K of the most popular nameservers being VPS instances hosted in AWS. Next, VERISIGN is rather expected, since it operates the gTLD nameservers. The world's 2 largest CDNs, AKAMAI and CLOUDFLARE together handle 13% of observed DNS traffic. However, since CLOUDFLARE generally uses IP anycast, we see a few times less nameserver IPs in the *servers* column than for AKAMAI. We also note substantially lower response delays and hop counts, which demonstrates that CDNs optimize these performance factors.

3.4 Query types

In Table 2, we analyze the top 10 QTYPEs, which in total represent 99.5% of all DNS transactions seen. The A query, used for fetching the IPv4 address, is clearly the dominant query type, being ≈3 times more popular than the AAAA query, used for fetching the IPv6 address (64% vs. 22% in *global*), which gives insight into the client-side adoption of IPv6 and its Happy Eyeballs algorithm (which we describe and analyze in depth in section 5). Moreover, while only 0.6% of responses to A queries neither contain the answer nor a delegation, for AAAA queries we see >40 times more NoData responses (25%), which in turn gives insight into the server-side IPv6 adoption, and into the issue of low negative caching TTLs (see section 5). Also note the 22% rate of NXDOMAIN responses for A queries vs. the 5.9% for AAAA: the data suggests A queries are more likely to be used for DNS scanning and other automated activity, such as Pseudo-Randomized Subdomain (PRSD) attacks [23].

We were surprised to learn that PTR is so popular, accounting for 6.4% of the DNS traffic. These queries are not normally issued by web users, and rather come from servers and other Internet infrastructure. The higher value for *qdots* of 6.8 was expected, given that reverse DNS records for IPv4 and IPv6 addresses have many labels (resulting in 6 and 34 dots, respectively). However, we saw 25 TLDs per minute, which demonstrates some use of PTR beyond reverse DNS [11] (normally only under the .arpa TLD). Note the highest among other QTYPEs response delay of 44 ms, which suggests that reverse DNS queries take ≈2 times longer than forward DNS queries.

Only 1.4% of all transactions seen were NS queries, of which a staggering 86% resulted in an NXDOMAIN response. By comparing in the *valid* column the number of *existing* FQDNs seen per minute vs. *all* FQDNs seen, we speculate that this is an indicator of ongoing PRSD attacks. In this context, note the substantially higher response size of 835 bytes, which puts additional strain on the DNS infrastructure.

We found TXT transactions to be as popular as NS transactions, yet surprisingly—comparing with other QTYPEs—the nameservers handling TXT queries are more distant from the resolvers (the value of *hops* equals to 10). Given the high value of *qdots*, the average number of FQDNs per SLD, and generally low TTL value of 5 seconds, we found the TXT records are often used for implementing custom, proprietary protocols over the DNS. More careful inspection revealed the origin of these queries in anti-virus and anti-spam systems.

3.5 Response delays

In Figure 3, we analyze response delays, i.e., the time difference between sending the DNS query at the recursive to the authoritative and receiving the response back at the recursive. Note that this delay generally comes from two sources: the Internet transmission delay, and the server processing delay. In order to better understand the possible network delay, we additionally consider the number of network hops between resolvers and nameservers, derived from the IP initial TTL value, e.g., [39]. Our assumption is that *in general* nameservers closer to resolvers (in terms of hop count) will have lower network delay, and thus should respond faster.

We use the Top-100K nameserver list, already characterized in subsection 3.2. On that list, only <0.9% of nameservers had the median response delay above 1 second. Thus, we plot the distribution of response delays in the range of 1-1000ms in Figure 3 (a). Here, we see roughly 4 sections of the CDF curve, marked with blue lines for readability: 1) consistent 0-5ms delays (3.1% of nameservers), presumably where the nameserver is co-located with the resolver within the same or adjacent AS (a common practice for large CDNs); 2) 5-35ms delays (22.3% of nameservers), presumably where the nameserver is located in the same or neighboring country; 3) 35-350ms (71.5% of nameservers), presumably where the nameserver is at distant location; 4) over 350ms (2.3% of nameservers), presumably where the nameserver or its Internet connectivity is impaired.

Next, we try to answer the question if the world's most popular nameservers are faster than the rest. In Figure 3 (b), we plot delays and hop counts for the top 25K nameservers: for readability, we present the data in groups of 100 neighboring nameservers, using their mean values. We find a pattern especially visible for the top 10K nameservers (which handle >50% of all DNS transactions, see subsection 3.2): lower response delays indeed seem correlated with the nameserver popularity. Moreover, the hop count statistic hints at nameservers simply being located closer to the resolvers.

Finally, in Figure 3 (c) and (d), we characterize the root and gTLD nameservers, respectively. In each case, there are 13 nameservers labeled 'A' through 'M', each deployed using IP anycast. We find varying median response delays for the root nameservers—yet roughly correlated with the hop count—which reflects the diversity in organizations and deployments behind the root nameservers. As expected, root nameservers with the most mirrors at different locations, namely E, F, and L, are also the fastest. In total, the root nameservers handle 3.0% of all DNS queries seen, 96.2% of which, however, result in an NXDOMAIN response [9].

The performance of gTLD nameservers is more consistent, revealing groups of servers with similar characteristics, and thus likely deployed in a similar way and at close locations. The B gTLD nameserver is the fastest, and thus the most affected by the botnet traffic described in subsection 3.2. In total, gTLD nameservers handle 9.6% of all DNS queries seen, 26.4% of which resulted in an NXDOMAIN response.

3.6 Use of QNAME minimization

As queried domains might reveal private information, the technique QNAME minimization (qmin) thwarts this by no longer sending the full original query name (QNAME) to the authoritative name servers. Instead, resolvers iteratively query longer QNAMEs by prepending labels from the original QNAME. As qmin was standardized relatively recently in 2016 [8], we evaluate its deployment on root and TLD name servers and compare our results to related work [13].

First, we group the authoritatives into root, TLD, and others using root zone data [38]. We evaluate the QNAMEs sent between each resolver and authoritative pair, as shown in Table 3. To increase confidence in our evaluation we only provide negative qmin results, i.e., we classify name servers as non-qmin instead of positively marking them as qmin. Root name servers are authoritative for the root zone. They are therefore expected to receive queries with only

	QTYPE	global	data	nodata	nxd	err	qdots	TLDs	eSLDs	FQDNs	valid	TTL	servers	delay	hops	size
1	A	**64%**	67%	**0.6%**	**22%**	11%	3.4	709	414,164	1,021,765	39%	60	105,422	20	7.5	121
2	AAAA	**22%**	57%	**25%**	**5.9%**	11%	3.5	623	213,694	528,504	80%	300	59,568	21	7.5	114
3	PTR	6.4%	45%	0.2%	29%	26%	**6.8**	**25**	363	144,283	54%	86400	22,414	**44**	8.2	129
4	NS	1.4%	9.4%	1.4%	**86%**	3.2%	2.4	149	5,169	6,470	**5.3%**	86400	3,859	22	8.0	**835**
5	TXT	1.4%	65%	4.1%	22%	8.1%	**5.9**	226	13,510	67,056	73%	**5**	7,548	40	**10**	118
6	MX	1.2%	60%	3.3%	2.9%	**34%**	2.6	255	33,390	39,686	86%	3600	13,630	29	7.7	113
7	SRV	1.1%	17%	3.4%	53%	27%	**6.8**	122	3,603	9,522	**22%**	300	8,540	25	7.6	137
8	CNAME	1.0%	28%	8.9%	**54%**	8.9%	4.4	192	**8,188**	**28,002**	35%	300	2,778	36	9.0	131
9	SOA	0.5%	40%	1.3%	39%	20%	**4.9**	101	9,843	10,564	46%	3600	5,149	24	7.2	128
10	DS	0.5%	43%	28%	28%	1.1%	2.6	247	20,617	23,688	69%	86400	1,037	**9.4**	7.1	**763**

Table 2: Top 10 QTYPEs: *global*) share in all observed DNS transactions; *data, nodata, nxd, err*) respectively, share of NoError+data, NoData, NXDOMAIN, and other errors in given QTYPE; *qdots*) number of QNAME labels; *TLDs, eSLDs FQDNs*) unique TLDs, effective SLDs, and FQDNs seen in NoError; *valid*) share of existing FQDNs; *TTL*) top TTL; *servers*) unique nameserver IPs; *delay, hops, size*) response delay [ms], network hops, and the response size [B]. Average values of 1-minute measurement windows (see section 2). Values highlighted in red are analyzed in subsection 3.4.

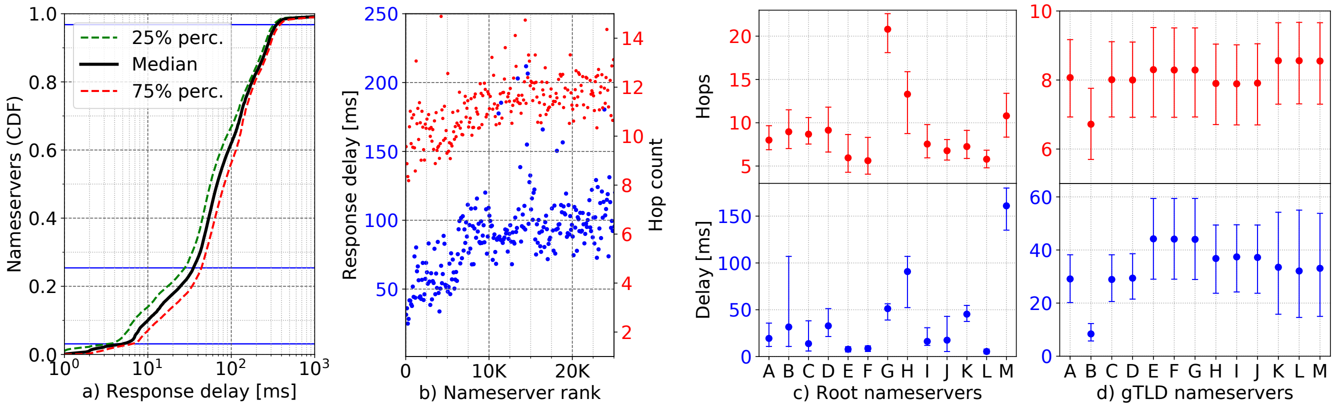

Figure 3: Response delays and number of network hops between DNS resolvers and nameservers: *a)* distribution of delays for Top-100K nameservers; *b)* patterns for Top-25K nameservers, each dot representing a group of 100; *c)* and *d)* median and quartiles for the root and gTLD nameservers, respectively (IPv4 anycast addresses).

one QNAME label (e.g., com) from qmin resolvers. If we find any QNAMEs in a resolver-root name server pair with more than one QNAME label (e.g., example.com), we classify this pair as non-qmin. TLD name servers are authoritative for the respective TLD zone. They are therefore expected to receive queries with at most two QNAME labels (e.g., example.com) from qmin resolvers. If we find any QNAMEs in a resolver-TLD name server pair with more than two QNAME labels (e.g., www.example.com), we classify this pair as non-qmin. In this study we only evaluate QNAMEs sent to root and TLD authoritatives, as other authoritatives (e.g., SLD name servers) can not be reliably classified as qmin or non-qmin.

When evaluating resolver-root name server pairs we find three resolvers which are possible qmin-enabled resolvers. All three resolvers are located in prefixes belonging to the same university and send DNS requests containing FQDNs with at most one QNAME label to different root servers. We check if these resolvers indicate non-qmin behavior for other name servers, which we did not find.

For resolver-TLD name server pairs we find two potential instances of qmin deployment. We again cross-check these resolvers with other authoritatives and do not find any indication that these resolvers employ non-qmin behavior towards them, as they only send QNAMEs with at most 2 labels to TLD servers. The resolvers belong to an IT business and to the same university as for the root server analysis.

Sent QNAME	Root NS	TLD NS	Other NS
com	?	?	?
example.com	✗	?	?
www.example.com	✗	✗	?

Table 3: Detecting resolvers not adhering to QNAME minimization (qmin) by inspecting their query behavior for root, TLD, and other authoritatives. '?' means that we can not determine the qmin status, '✗' means that the resolver does not employ qmin.

We also make the requirement of qdots more lenient to account for TLD name servers which host zones with more than one labels (e.g.,.uk also hosts .co.uk, .il also hosts .org.il, .me also hosts .net.me). We whitelist these TLD name server configurations to allow QNAMEs with three labels. This less strict definition of qmin, however, does not find any additional qmin resolvers.

In total qmin-enabled resolvers send about 18 k queries to root name servers and 4 k to TLD name servers per day. This makes up a minuscule share of 0.005 % and 0.0001 % of all traffic to root and TLD servers, respectively. This is in stark contrast to related work which sees about 40 % qmin deployment at the .nl TLD name server [13]. A more recent study by Huston [34] also finds low deployment of qmin, finding that about 3 % of users take advantage of the technique. One possible explanation for varying deployment statistics is the different methodology used to identify qmin queries, i.e., DeVries *et al.* label a resolver as qmin if 97 % of queries are classified as minimized [13], whereas we apply a strict notion of 100 % qmim queries.

3.7 Data representativeness

We run several experiments to evaluate the representativeness of our data. First, in Figure 4a), we show that the number of authoritative nameservers seen in 1 hour converges to a limit of 500K-600K when we increase the fraction of available vantage points from 0 to 100%. In each step, we take a random sample of all available resolvers, and listen to DNS traffic continuously for 1 hour. We report the average values obtained by repeating the experiment 20 times. The plot suggests we miss some visibility beyond the 100% mark (which represents all resolvers used for this paper), but the shape of the curve suggests a relatively small, bounded number of the "missing" nameservers. Otherwise, if the number of resolvers was too low to characterize the DNS nameservers, we would see a curve not converging to a limit, or even a linear relation.

In order to preview anticipated improvements to our system, the blue curves labelled "DNS Observatory" show the results obtained for this paper (using only the main SIE passive DNS channel), whereas the red curves labelled "Available data" show the results obtained using more vantage points (using all available SIE passive DNS channels), planned for future inclusion.

Recall that in Figure 2a) we demonstrate that the majority of the observed DNS traffic is handled by a small number of the most popular nameservers. Thus, in this experiment, the new nameservers becoming visible by increasing the sample size are likely unpopular, carrying diminishing amounts of DNS traffic. In other words, the big nameservers are already well visible through a small set of resolvers, smaller than what we used for this paper. We demonstrate this in Figure 4b), where we plot the fraction of an hourly Top-10K nameserver list visible using subsets of our resolver pool: even a 5% sample is enough to see 95% of the list.

In Figure 4c), we present a similar experiment where we plot the number of Top-Level Domains seen in 1 hour as a function of the fraction of recursive resolvers used for monitoring the DNS. Although there are over 1,500 TLDs existing in the DNS [37], we stress not all of them are actively used on the Internet, hence the limit of 1,150 TLDs that our curves converge to. Adding the resolvers from the other SIE channels (red curve) does not bring us much more coverage, which suggests DNS Observatory already has decent visibility into various TLDs.

Next, in a similar manner as above, we validate that the other results presented in this paper—concerning the response delays, number of router hops, QTYPEs, TTLs and their estimated probabilities—all converge to the reported values as we increase the fraction of used vantage points (we skip the plots for brevity).

Finally, although we deliberately avoided tracking less popular nameservers in the DNS, we want to better understand our data and what we skipped in it. In Figure 5, we plot the number of *all* seen nameserver IP addresses as a function of monitoring time. That is, here we use all vantage points, and in each step we increase by 1h for how long we continuously record the raw traffic. In total, over 3 days (instead of 1 hour used above), we observed 1.5M unique IPs of authoritative nameservers *in active use*. The IPv4 addresses were contained within 405 k /24 prefixes, yet 48 % of these observed prefixes had only 1 address, 24 % had 2 addresses, and 7.7 % had 3 addresses. This means that the less popular nameservers that we intentionally skipped in subsection 3.2 are actually well distributed on the IP addressing space. For completeness, we visualize this with a heatmap in Figure 6.

DNS Observatory is an ongoing project aimed at gaining a bird's-eye view on the DNS. In this work, we evaluate our novel system using only the main SIE passive DNS channel, due to limited processing bandwidth. More work is planned in the near future to ingest all of the SIE passive DNS channels, which will improve data coverage in terms of resolver count and their geographical locations. Due to confidentiality agreements with the SIE data contributors (our data comes from a strict security context) we must not disclose their identities or locations. However, note that access to SIE is available to other researchers via data sharing and through a research grant program, which allows for independent verification of our results [19, 21, 24].

4 UNDERSTANDING THE DNS TTL

DNS responses are accompanied by a TTL value, which is a 32 bit field that sets an upper bound on how long a resource record can be kept in the cache [47]. Previous studies have shown the role of caching in DNS resilience against DDoS attacks [48, 52]. In addition, there have been evaluations on the effectiveness of caching and its performance implications [41] and efforts to build models for TTL caches [40] (see section 6).

In the following, (1) we analyze the effect of TTL changes on traffic between recursive and authoritative nameservers and (2) we assess the feasibility of leveraging TTL information to detect upcoming DNS infrastructure changes.

4.1 How TTLs affect traffic volumes?

The duration for how long a record can be cached—determined by its TTL—influences the number of queries. Below, we put the notion of a TTL decrease leading to a query increase to the test.

We evaluate SLD data from March and April 2019 to find large TTL changes between these two months. In Figure 7, we depict the case of the `xmsecu.com` SLD, which slashed its TTL from 10 minutes to 10 seconds on April 4, 2019. This domain provides a web interface for infamous Xiongmai video surveillance devices,

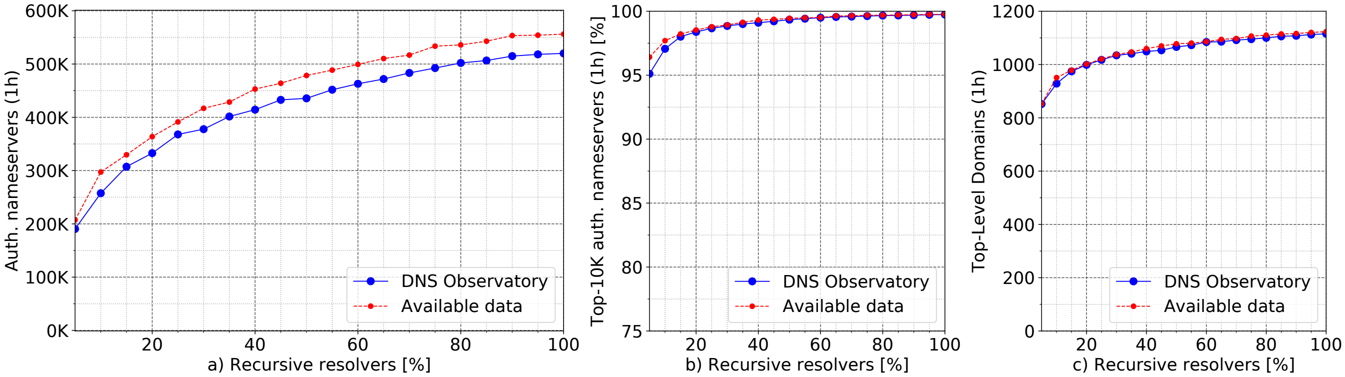

Figure 4: Evaluating data representativeness: a) authoritative nameserver IP addresses seen in 1 hour as a function of an increasing sample of recursive resolvers (average for 20 repetitions); b) fraction of Top-1K nameservers seen in 1 hour; c) number of observed TLDs in 1 hour. "Available data" show planned improvements.

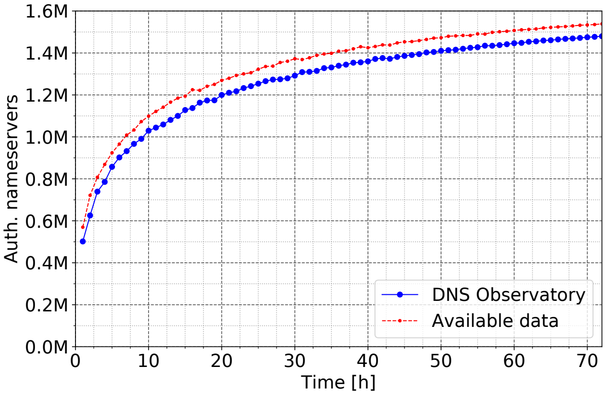

Figure 5: Number of all observed authoritative nameserver IP addresses as a function of time, using all vantage points.

which were hacked by the Mirai botnet [66]. In addition to their involuntary participation in DDoS botnets, they seem to voluntarily put an additional load on the TLD nameservers. This SLD shows a clear relation between TTL decrease and DNS query increase.

In Figure 8, we depict the top 100 SLDs by traffic change between March and April 2019. Intuitively, the majority of TTL decreases on the left side of the plot lead to an increase in traffic. With some exceptions, we see an almost inverse linear correlation between TTL and DNS traffic changes.

On the right side of Figure 8, the case is not as clear cut. Even though we see quite a few cases of query rate decreases which could be correlated to DNS TTL changes, there are twice as many SLDs with increased (34 cases) than decreased traffic (17 cases) even though their TTL goes up. We investigate these seemingly inconsistent cases and find that 28 of the 34 cases only increase their query rate, but not their response rate, i.e., resolvers are increasingly querying for non-existent FQDNs or issuing otherwise unusual queries.

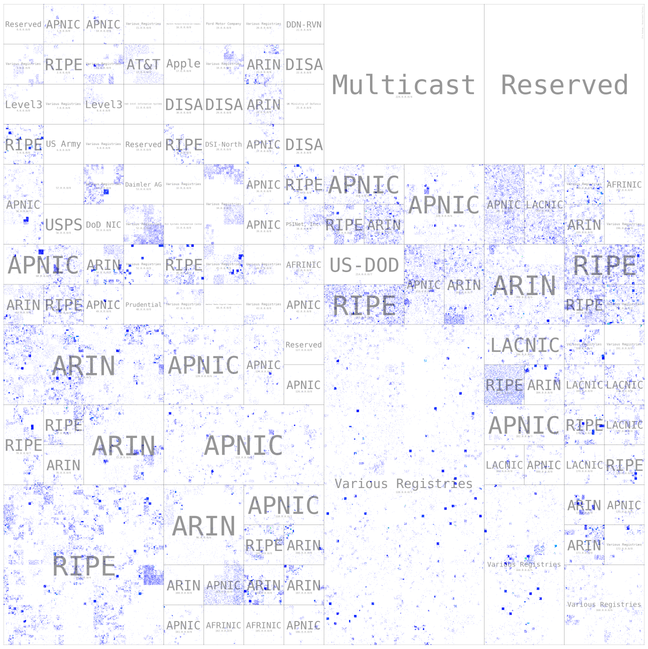

Figure 6: Hilbert space-filling curve heatmap of all observed IPv4 addresses of authoritative nameservers (produced using [68]). Each pixel corresponds to a /24 prefix. The blue color means 1 address in given prefix used as a nameserver during a 3-day time window.

To summarize, we find that for the most part TTLs directly influence query rates, as caching a name for a shorter duration leads to more queries. We detect inconsistent behavior by many SLDs with an increased TTL, which can be attributed to NXDOMAIN queries, or simply the domain becoming more popular in the meantime.

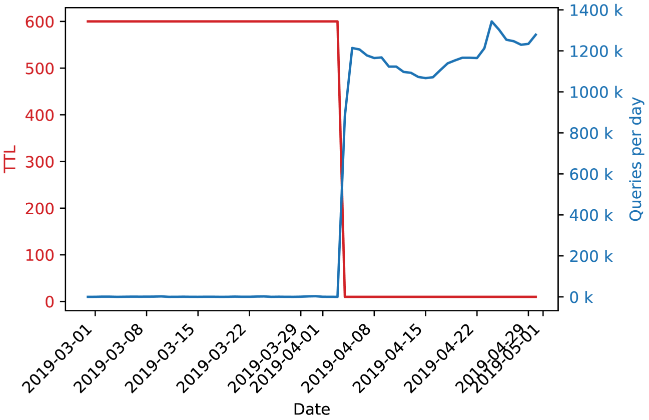

Figure 7: TTL decrease from 600 to 10 seconds on `xmsecu.com` leading to a massive increase in queries.

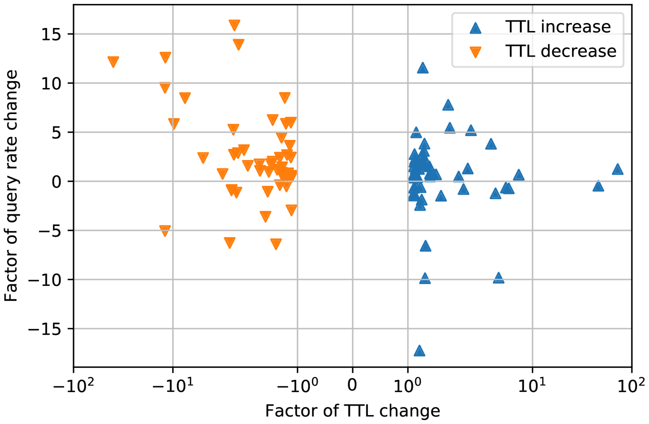

Figure 8: Correlating TTL increases and decrease to changes in number of queries. Note that the y-axis is symlog-scaled for improved readability.

4.2 Can TTL changes indicate changes in DNS infrastructure?

While there is no consensus on how to choose TTL values for different types of resource records—e.g., CDNs are known for using values in the range of minutes, while for the root zone it is of the order of days—operators decrease TTL values of records before carrying out changes in its associated infrastructure [43].

For example, consider the domain name `example.org`, which has two nameservers `[a,b].iana-servers.net` with a TTL of 86400s (1 day). Now lets assume its operator decides to change the DNS provider, and use different NS records of `ns[a,b].example.org`. Before carrying out this change, the operator of `example.org` may reduce the TTL of its NS records from 1 day to, e.g., 30s, and only then update the NS records—i.e., after waiting a time interval long enough so that the previously cached records expire at all resolvers. Any potential issues with the change can be resolved by quickly reverting to the "old" NS records. Once the operation is successfully performed, the operator may then increase the TTL values of the new NS records. Below, we investigate if we can confirm these reports.

4.2.1 Methodology. We use the **aafqdn** dataset described in subsection 3.1. We collect only the NoError responses that either have a non-empty ANSWER section or contain NS records in the AUTHORITY section (or both). Note that a DNS response may be composed of up to three sections: answer , authority, and additional sections [47]. Moreover, we consider only the responses that come from authoritative nameservers—i.e., from child delegations—which have the AA flag set. Each hour, we produce a data file with Top-10K FQDNs, as described in subsection 2.4.

We analyze consecutive hourly files from April 19th until 25th, 2019. For each FQDN, we analyze the TTL distribution of its A and NS records, and detect changes in these values. We classify a change in TTL values if at least 10% of the DNS responses for the particular hour show new values. Notice that the top list of each hourly file may differ, given they are dictated by how popular the FQDNs were on that specific hour, due to user interest and diurnal patterns [55].

For the aforementioned period, we found 65 domains with significant TTL changes in their records. We classify them into categories in Table 4. In order to better understand the changes detected using DNS Observatory, we manually lookup the FQDNs in DNSDB [18], which provides a more detailed, historical record of the DNS.

4.2.2 Analysis. Table 4 shows the results of our classification. The most common category is non-conforming authoritative name servers returning variable TTLs for the same domain in subsequent queries. For example, the authoritative server dns.widhost.net is authoritative for dns2.vicovoip.it, and when asked directly for its A record, answers with decreasing TTL values (<1024). While these TTL changes do not indicate changes in the infrastructure of the domain name (thus misguiding our detection), it indicates that some domains do not conform to the standard behavior of returning the same TTL values *unless* there are zone file updates.

Another common category found associated with TTL changes are renumbering events, i.e., changing either A or AAAA records of domains. We can see that ns2.oh-isp.com was renumbered to 52.166.106.97. Checking `whois`, we see the new A record belongs to Microsoft, suggesting that this DNS server is now hosted in a cloud and—when the change was completed—the TTL increased from 600s to 38400s.

We also saw four changes in TTL not associated with any change in DNS infrastructure (TTL Decrease/Increase in Table 4). These events show simple increase or decrease of TTLs for domains, and not necessarily renumbering or changes in NS records.

Moreover, our method allows to spot domains that change NS and A records at the same time (Change NS), which were accompanied by TTL reductions from 600 to 10s. We could not, however, classify 21 changes given there was not enough data in DNSDB to understand these results.

Overall, our method allows us to detect various types of changes to the DNS infrastructure, or domains that have non-conforming responses. Given that DNS Observatory covers the DNS beyond SLDs, it can be used to track changes to domains at any level of the DNS hierarchy. Thus, we confirm the changes of TTLs can be used

Category	#	Type	Example	TTL before/after	Change	Date Change	Comment
Non-conforming	17	A	dns2.vicovoip.it	variable TTL	NA	2019-04-23 01:00	Dynamic TTL
Renumbering	13	A	ns2.oh-isp.com	600/38400	31.222.208.197 → 52.166.106.97	2019-04-23 10:27	Change to MS cloud
		A	kaitest.stou2.com	300/60	104.31.11[4,5].142 → 104.31.13[8,9].10	2019-04-21 19:18	–
TTL Decrease	3	A/NS	ns2.mtnbusiness.co.ke	86400/3600	None	2019-04-24 01:00	–
TTL Increase	1	A	ns2.whiteniledns.net.	120/300	None	2019-04-25 04:00	–
Change NS	1	NS/A	jia003.top.	600/10	f1g1ns[1,2].dnspod.net → ns[3,4].dnsv2.com	2019-04-21 07:30	Change NS and A
Unknown	21	NS	u1.hoster.by	3600/300	Unknown	2019-04-22 09:00	–

Table 4: TTL changes detected and classification

to *predict* changes to the DNS infrastructure—even if they do not occur that often.

5 HAPPY EYEBALLS AND NEGATIVE CACHING TTLS

Another interesting effect we can observe with the data from DNS Observatory is the impact of the devices using the Happy Eyeballs algorithm on the volume of empty DNS responses, due to low negative caching TTLs.

5.1 Background

Happy Eyeballs is a standardized algorithm [61, 69] where a host with IPv4 and IPv6 connectivity sends both A and AAAA queries to learn IPv4 as well as IPv6 addresses of a domain name. The host then tries to establish a connection to the returned IP addresses—normally a slight advantage of a few ms is given to IPv6. For domains that only map to an IPv4 address and no IPv6 address, the use of Happy Eyeballs expectedly leads to empty AAAA responses (i.e., NoData). These empty AAAA responses should be cached by the resolver, in order to avoid constant resending of the same AAAA queries for domain names without associated IPv6 addresses. The caching duration is determined by the domain’s SOA record: its rightmost value specifies the negative caching TTL.

We acknowledge that the TTLs are not the only determining factor influencing the frequency of queries for particular domain—resolver configuration, query prefetching, and specific implementation details can play an important role. However, lower negative caching TTLs generally lead to higher DNS traffic volumes. Note that, according to [53], DNS is already the top producer of new flows in many subscriber networks.

For IPv4-only domains, if the negative caching TTL is much lower compared with the regular A record TTL, this leads to many of *all* DNS responses being empty AAAA responses. This DNS traffic of questionable usefulness increases the load on authoritative nameservers and also causes more query sending and processing by recursive resolvers. Moreover, note the “Resolution Delay” of the Happy Eyeballs v2 algorithm [61], which by default makes the device wait up to 50ms for the response to AAAA query, even if the A response is received quicker. Thus, the additional time needed to query the authoritative nameserver instead of using the resolver cache will be directly reflected in the IPv4 connection delay.

5.2 Correlating low negative caching TTL with empty AAAA responses

In the following, we analyze the top 200 FQDNs by DNS traffic in April 2019.

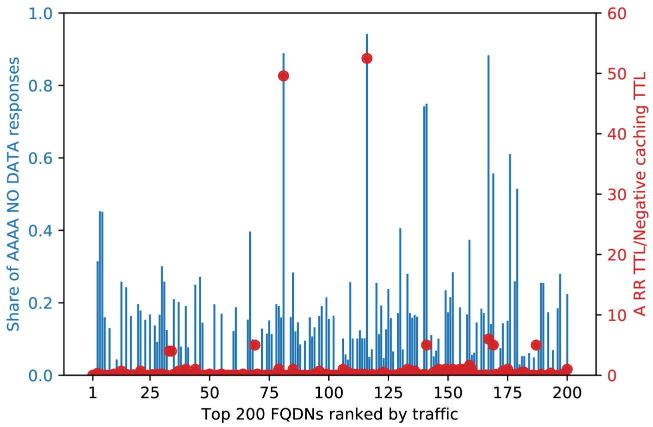

Figure 9: Comparing negative caching TTLs to share of AAAA NO DATA responses for top 200 FQDNs ranked by traffic. On the left y-axis we show the share of empty AAAA responses for each specific top-200 FQDN. The right y-axis displays the quotient of the A resource TTL by the negative caching TTL. The larger the quotient the more likely many empty AAAA responses.

Figure 9 shows the impact of low negative caching TTLs in combination with the Happy Eyeballs algorithm on the number of empty AAAA responses. In the top 200 FQDNs, we find 5 FQDNs with more than 70 % of all responses being empty AAAA responses. These empty responses and the triggering queries lead to an increased load at the resolvers and authoritative nameservers.

Most striking are the two FQDNs at ranks 81 and 116, which are used for network time services of a popular operating system. Both domains have a regular A record TTL between 10 and 15 minutes, but a 50 times lower negative caching TTL of 15 *seconds*. This discrepancy leads to 89 % and 94 % of all responses being AAAA NoData, respectively.

The FQDN at rank 141, which belongs to a large ad network, has a five times shorter negative caching TTL compared to the A TTL, i.e., 300 vs. 60 seconds. This in turn leads to a share of 75 % of empty AAAA respones.

Similarly, at rank 167 we find an FQDN used for operating system updates hosted by a large CDN with an A record TTL of 1 hour, but a negative caching TTL of just 10 minutes. This six times lower negative caching TTL in combination with the Happy Eyeballs algorithm leads to 88 % of all DNS responses being empty AAAA responses.

Interestingly, at rank 140 we see an example of an FQDN, used for hosting blogs, with more than 74 % of empty AAAA responses, but an actually six times *higher* negative caching TTL compared with the A record TTL. We suspect that this artifact is due to some resolvers not respecting its relatively high negative caching TTL of 1 hour.

We notified operators of the nameservers most affected by the use of low negative caching TTLs. We received a response from a large CDN stating that they use low negative caching TTLs purely as a defensive measure, in case of an error in the way they generate dynamic authoritative responses.

5.3 Effect of deploying IPv6

In the light of Happy Eyeballs leading to many AAAA queries and subsequent empty responses for non-IPv6 FQDNs, we investigate FQDNs that enabled IPv6 during our observation period in April 2019.

We find 10 FQDNs that added IPv6 support and started sending AAAA responses containing actual data. By analyzing each FQDN we see the number of empty AAAA responses going down after IPv6 activation as expected. Additionally, we find no significant change in query volume correlated to the activation of IPv6 for each FQDN. This is not surprising, as all 10 FQDNs use equal or larger negative caching TTLs compared with regular TTLs.

We conclude that activating IPv6 for FQDNs does not significantly change query volumes, if the negative caching TTLs are similar to regular TTLs. If the negative caching TTLs are much shorter than regular TTLs, this changes as shown in subsection 5.2.

5.4 Discussion

With the continuously growing IPv6 deployment [28] and clients using the Happy Eyeballs algorithm [61, 69] to learn IPv4 and IPv6 addresses for each domain name, authoritative servers need to cope with an increasing query load. As shown in section 5 especially the combination of IPv4-only domains and low negative caching TTLs lead to many empty AAAA responses.

Three possible courses of action are (1) adding a new query type for joint A and AAAA query, (2) separating the negative caching between NXDOMAIN and NODATA, or (3) simply changing the negative caching TTL to a value more in line with the TTL of A or other query types.

The first option would enable clients to request IPv4 and IPv6 addresses for the same query name. Although this is similar to an ANY query, the response would be limited to A and AAAA records. In addition, a single TTL for both IPv4 and IPv6 addresses could be used, making caching easier. If any server in the chain—i.e., recursive resolver or authoritative nameserver—does not support this new query type, the requesting client could fall back to sending separate A and AAAA queries.

The second option goes back to discussions with DNS operators. They suggested to split semantic of the negative caching TTL, which is currently used for NXDOMAIN as well as NODATA. As a result, zone administrators who are worried about NXDOMAIN resulting from zone misconfigurations could then choose a shorter NXDOMAIN TTL and a NODATA TTL resembling the regular A record TTL. Consequently, the share of empty AAAA responses could be greatly decreased.

The third option—setting the negative caching TTL to similar value as the A record TTL—is the simplest in terms of configuration effort, and requires no changes to the DNS itself. It has, however, drawbacks, as some DNS operators confirmed us that they are explicitly setting low negative caching TTLs as a defensive measure, reducing the time needed to distribute remediated zones through caches.

Generally, we want to emphasize that low negative caching TTLs in combination with the Happy Eyeballs have a direct negative impact on client latency, as they need to wait for the recursive resolver to issue AAAA queries instead of serving them from the cache.

6 RELATED WORK AND DATASETS

Active measurements of the DNS: the OpenINTEL project by Rijswijk-Deij *et al.* [65] is a large active DNS measurement study that makes daily scans of all domains under legacy gTLDs, new gTLDs, and a dozen ccTLDs, for a total of 218 million SLDs daily. Each domain is measured using many features, with an exceptional support for DNSSEC [51]. Comparing with our work, DNS Observatory provides a different view on the DNS, based on passive, aggregated measurements of the most popular DNS objects, such as authoritative nameserver IPs, SLDs, FQDNs, and more. Instead of actively sending 1 set of queries per SLD every 24 hours from 1 location, we extract our data from real-world DNS traffic collected globally. Thus, we are not limited to a fixed set of QTYPEs or TLDs, and we monitor the traffic volumes.

Project Sonar run by Rapid7 Labs provides the Forward DNS dataset [57] that contain answers to A, AAAA, ANY, CNAME, MX, and TXT lookups for many FQDNs collected by the project (including web crawling, reverse DNS, SSL certificates, TLD zone files [56]). Thus, it resembles the OpenINTEL project but collects no features (except for the returned records), scans the DNS less frequently, and in general has less SLD coverage.

The RIPE Atlas project [59], a global Internet measurement platform, publishes the data collected by its users, which includes DNS responses [58]. The system has over 10 k probes spread around the world that allow to run various active measurements of recursive and authoritative DNS servers. However, collected data depends on lists of domains queried by RIPE Atlas users, and thus data coverage compared with DNS Observatory or OpenINTEL is quite limited.

The above DNS studies and datasets are obtained using *active measurements*, i.e., the tools actively send DNS queries for predetermined lists of domains. In our paper, however, the data is obtained in *passive* measurements: we analyze traffic from production resolvers spread over multiple locations to authoritative nameservers. As such, our data is a direct function of user activity and cache misses, has finer granularity (multiple vantage points and time aggregations), and covers not only TLDs, but also SLDs, FQDNs, QTYPEs, and many other DNS objects. On the other hand, note that research on IPv6 addressing often relies on periodic large-scale DNS lookups—targeting e.g., zone files, Certificate Transparency domains, and domains learned from rDNS—in order to gather IPv6 addresses for scanning the Internet [26, 27].

Passive measurements of the DNS: In a 2012 paper [24]—updated in [25]—Gao *et al.* characterize the global DNS behavior using the same raw data source as DNS Observatory, but using an early version of the SIE and in a much shorter time frame: 2 weeks (26 billion transactions) vs. 4 months (1.6 trillion transactions). Their analyses are largely complementary to the study we presented, as this paper focus on authoritative nameservers instead of the resolvers. Moreover, DNS Observatory is designed for long-term and real-time operation, and provides insight into more areas of the DNS, e.g., popularity of IP addresses in DNS answers.

DNS-OARC provides the Day in The Life of the Internet (DITL) datasets [16] that record DNS traffic at many root and TLD nameservers for a few days each year. ICANN runs live DNS Stats at [36] that allows to visualize various statistics for many nameservers, foremost the L-root nameservers. ENTRADA [70] is an open-source platform for storing and processing DNS traffic recorded at authoritative nameservers, e.g., to visualize the .nl ccTLD statistics [62]. Similarly, Deri *et al.* describe another system for monitoring the .it ccTLD nameservers in [14]. Interestingly, Mark Allman publishes anonymized summaries of *below-recursive* lookups made by devices in the Case Connection Zone network [3] (∼100 homes in Cleveland, Ohio).

DNS caching, TTLs, and resilience: several research works investigated the relationship between TTLs and DNS caching. Given that we monitor only the *cache miss* queries, TTLs play an important role in our datasets. Jung *et al.* [41], using simulations based on real traces, showed that longer TTLs improve caching, mostly for domains with short TTLs (<1000s). In a subsequent study [40], Jung *et al.* modeled DNS caches and were able to predict cache hit rates from their previous study. Moura *et al.* [48] analyzed the relationship between TTL and DNS resilience in face of DDoS attacks. In 2007 Pappas *et al.* proposed changes to caching strategies for NS records to improve DNS resilience against DDoS attacks [52].

Query name minimization: RFC 7816 standardized query name minimization (qmin) in 2016 [8]. A few years later, Vries *et al.* presented the first study focusing on qmin deployment [13]: with a more local view on the DNS, they found about 40 % of all queries to the .nl TLD and K-root nameservers being sent with qmin. However, a more recent study by Huston [34] reported only 3 % deployment of qmin, which is closer to our results in subsection 3.6.

7 CONCLUSION

We presented DNS Observatory, a novel stream analytics platform that allows for unprecedented visibility into the DNS, and we demonstrated some of its capabilities on real-world data.

DNS Observatory is based on passive measurements and collects data from a large, diverse set of recursive resolvers spread around the world, and run by many different operators. Using various stream-oriented algorithms and probabilistic data structures, we were able to ingest and process a total of 1.6 trillion DNS resolver-nameserver transactions executed between January and April 2019. We obtained a bird's-eye view on the DNS, which allows for its better understanding, and which suggests possible improvements.

The aggregated datasets collected in DNS Observatory are already available to academic researchers, by contacting the primary author of this paper or through the Farsight Security Research Grant program [19]. In a longer perspective, we also plan to make parts of the collected data publicly available through a web interface, linked under https://www.farsightsecurity.com/.

ACKNOWLEDGEMENTS

We would like to thank people who reviewed a pre-print version of this paper for their great feedback that improved our work: anonymous IMC reviewers and our shepherd Alberto Dainotti, Joe St Sauver, Eric Ziegast, and Jeremy C. Reed. We also thank the IETF MAPRG chairs, Mirja Kühlewind and Dave Plonka, for the opportunity of presenting early results of DNS Observatory at IETF105 in Prague. Finally, we thank all SIE data contributors that run Farsight passive DNS sensors, who made this work possible.

REFERENCES

[1] J. Abley and K. Lindqvist. 2006. Operation of Anycast Services. RFC 4786 (Best Current Practice). , 24 pages. https://doi.org/10.17487/RFC4786

[2] Mark Allman. 2018. Comments On DNS Robustness. In *Proceedings of the Internet Measurement Conference 2018*. ACM, 84–90.

[3] Mark Allman. 2019. Case Connection Zone DNS Transactions. http://www.icir.org/mallman/data.html.

[4] Mark Allman and Vern Paxson. 2007. Issues and Etiquette Concerning Use of Shared Measurement Data. In *ACM Internet Measurement Conference*.

[5] Mario Almeida, Alessandro Finamore, Diego Perino, Narseo Vallina-Rodriguez, and Matteo Varvello. 2017. Dissecting DNS Stakeholders in Mobile Networks. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. ACM, 28–34.

[6] Amazon Route 53. 2019. Choosing a Routing Policy. https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html.

[7] Burton H. Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* (1970).

[8] S. Bortzmeyer. 2016. DNS Query Name Minimisation to Improve Privacy. RFC 7816 (Experimental). , 11 pages. https://doi.org/10.17487/RFC7816

[9] Sebastian Castro, Duane Wessels, Marina Fomenkov, and Kimberly Claffy. 2008. A day at the root of the internet. *ACM SIGCOMM Computer Communication Review* 38, 5 (2008), 41–46.

[10] Yizheng Chen, Manos Antonakakis, Roberto Perdisci, Yacin Nadji, David Dagon, and Wenke Lee. 2014. DNS noise: Measuring the pervasiveness of disposable domains in modern DNS traffic. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 598–609.

[11] S. Cheshire and M. Krochmal. 2013. DNS-Based Service Discovery. RFC 6763 (Proposed Standard). , 49 pages. https://doi.org/10.17487/RFC6763 Updated by RFC 8553.

[12] C. Contavalli, W. van der Gaast, D. Lawrence, and W. Kumari. 2016. Client Subnet in DNS Queries. RFC 7871 (Informational). , 30 pages. https://doi.org/10.17487/RFC7871

[13] Wouter B de Vries, Quirin Scheitle, Moritz Müller, Willem Toorop, Ralph Dolmans, and Roland van Rijswijk-Deij. 2019. A First Look at QNAME Minimization in the Domain Name System. In *International Conference on Passive and Active Network Measurement*. Springer, 147–160.

[14] Luca Deri, Lorenzo Luconi Trombacchi, Maurizio Martinelli, and Daniele Vannozzi. 2012. A Distributed DNS Traffic Monitoring System. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 30–35.

[15] David Dittrich et al. 2012. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *US DHS* (2012).

[16] DNS-OARC, CAIDA, ISC. 2018. A Day in the Life of the Internet (DITL). https://www.dns-oarc.net/oarc/data/ditl.

[17] D. Eastlake 3rd and M. Andrews. 2016. Domain Name System (DNS) Cookies. RFC 7873 (Proposed Standard). , 25 pages. https://doi.org/10.17487/RFC7873

[18] Farsight Security. 2019. DNSDB. https://www.farsightsecurity.com/solutions/dnsdb/.

[19] Farsight Security. 2019. Farsight Grant Programs. https://www.farsightsecurity.com/grant-access/.

[20] Farsight Security. 2019. Passive DNS Sensor. https://www.farsightsecurity.com/technical/passive-dns/passive-dns-sensor/.

[21] Farsight Security. 2019. SIE Data Sharing. https://www.farsightsecurity.com/community/data-sharing/.

[22] Farsight Security. 2019. SIE: Security Information Exchange. https://www.farsightsecurity.com/solutions/security-information-exchange/.

[23] Shir Landau Feibish, Yehuda Afek, Anat Bremler-Barr, Edith Cohen, and Michal Shagam. 2017. Mitigating DNS random subdomain DDoS attacks by distinct

heavy hitters sketches. In *Proceedings of the fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies*. ACM, 8.

[24] Hongyu Gao, Vinod Yegneswaran, Yan Chen, Phillip Porras, Shalini Ghosh, Jian Jiang, and Haixin Duan. 2013. An empirical reexamination of global DNS behavior. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. ACM, 267–278.

[25] Hongyu Gao, Vinod Yegneswaran, Jian Jiang, Yan Chen, Phillip Porras, Shalini Ghosh, and Haixin Duan. 2016. Reexamining DNS from a global recursive resolver perspective. *IEEE/ACM Transactions on Networking (TON)* 24, 1 (2016), 43–57.

[26] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *Proceedings of the 2018 Internet Measurement Conference*. ACM, New York, NY, USA, 15. https://doi.org/10.1145/3278532.3278564

[27] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. In *TMA*.

[28] Google. 2019. IPv6 Adoption. https://www.google.com/intl/en/ipv6/statistics.html.

[29] Google. 2019. Protocol Buffers. https://developers.google.com/protocol-buffers/.

[30] Stefan Heule, Marc Nunkesser, and Alex Hall. 2013. HyperLogLog in Practice: Algorithmic Engineering of a State of The Art Cardinality Estimation Algorithm. In *Proceedings of the EDBT 2013 Conference*. Genoa, Italy.

[31] Hubert, Bert. 2019. Herding the DNS Camel. https://www.ietf.org/blog/herding-dns-camel/.

[32] Hubert, Bert. 2019. The DNS Camel. https://powerdns.org/dns-camel/.

[33] Hubert, Bert. 2019. The DNS Camel... https://blog.apnic.net/2018/03/29/the-dns-camel/.

[34] Geoff Huston. 2019. DNS Query Privacy. https://blog.apnic.net/2019/08/12/dns-query-privacy/.

[35] Huston, Geoff. 2019. AS Names. https://www.potaroo.net/.

[36] ICANN. 2019. DNS Stats. http://stats.dns.icann.org/.

[37] ICANN. 2019. List of Top-Level Domains. https://www.icann.org/resources/pages/tlds-2012-02-25-en.

[38] InterNIC. 2019. Root zone data. https://www.internic.net/domain/root.zone.

[39] Cheng Jin, Haining Wang, and Kang G Shin. 2003. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 30–41.

[40] Jaeyeon Jung, Arthur W. Berger, and Hari Balakrishnan. 2003. Modeling TTL-based Internet Caches. San Francisco, CA, USA. http://www.ieee-infocom.org/2003/papers/11_01.PDF

[41] Jaeyeon Jung, E. Sit, H. Balakrishnan, and R. Morris. 2002. DNS performance and the effectiveness of caching. *IEEE/ACM Transactions on Networking* 10, 5 (Oct 2002), 589–603. https://doi.org/10.1109/TNET.2002.803905

[42] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: a research-oriented top sites ranking hardened against manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. Internet Society.

[43] Cricket Liu and Paul Albitz. 2006. *DNS and BIND* (5 ed.). O'Reilly Media, Inc., 192–194.

[44] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. 2005. Efficient computation of frequent and top-k elements in data streams. In *International Conference on Database Theory*. Springer, 398–412.

[45] P.V. Mockapetris. 1983. Domain names: Concepts and facilities. RFC 882. , 31 pages. https://doi.org/10.17487/RFC0882 Obsoleted by RFCs 1034, 1035, updated by RFC 973.

[46] P.V. Mockapetris. 1983. Domain names: Implementation specification. RFC 883. , 74 pages. https://doi.org/10.17487/RFC0883 Obsoleted by RFCs 1034, 1035, updated by RFC 973.

[47] P.V. Mockapetris. 1987. Domain names - concepts and facilities. RFC 1034 (Internet Standard). , 55 pages. https://doi.org/10.17487/RFC1034 Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020, 8482.

[48] Giovane C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt, and Marco Davids. 2018. When the Dike Breaks: Dissecting DNS Defenses During DDoS. https://doi.org/10.1145/3278532.3278534

[49] Mozilla Foundation. 2019. Public Suffix List. https://publicsuffix.org/.

[50] Nipravsky, Tom. 2018. Meet MyloBot – A New Highly Sophisticated Never-Seen-Before Botnet That's Out In The Wild. https://www.deepinstinct.com/2018/06/20/meet-mylobot-a-new-highly-sophisticated-never-seen-before-botnet-thats-out-in-the-wild/.

[51] OpenINTEL. 2019. Data dictionary. https://openintel.nl/background/dictionary/.

[52] V. Pappas, D. Massey, and L. Zhang. 2007. Enhancing DNS Resilience against Denial of Service Attacks. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*. 450–459. https://doi.org/10.1109/DSN.2007.42

[53] David Pariag and Tim Brecht. 2017. Application bandwidth and flow rates from 3 trillion flows across 45 carrier networks. In *International Conference on Passive and Active Network Measurement*. Springer, 129–141.

[54] Craig Partridge and Mark Allman. 2016. Ethical Considerations in Network Measurement Papers. *Commun. ACM* (2016).

[55] Lin Quan, John Heidemann, and Yuri Pradkin. 2014. When the Internet Sleeps: Correlating Diurnal Networks With External Factors. In *Proceedings of the ACM Internet Measurement Conference*. ACM, Vancouver, BC, Canada, 87–100. https://doi.org/10.1145/2663716.2663721

[56] Rapid7 Labs. 2019. Forward DNS: data sources. https://github.com/rapid7/sonar/wiki/Forward-DNS.

[57] Rapid7 Labs. 2019. Project Sonar: Forward DNS (FDNS). https://opendata.rapid7.com/sonar.fdns_v2/.

[58] RIPE Atlas. 2019. Measurements. https://atlas.ripe.net/measurements/.

[59] RIPE NCC Staff. 2015. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal* 18, 3 (2015).

[60] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D Strowes, and Narseo Vallina-Rodriguez. 2018. A long way to the top: significance, structure, and stability of internet top lists. In *Proceedings of the Internet Measurement Conference 2018*. ACM, 478–493.

[61] D. Schinazi and T. Pauly. 2017. Happy Eyeballs Version 2: Better Connectivity Using Concurrency. RFC 8305 (Proposed Standard). , 15 pages. https://doi.org/10.17487/RFC8305

[62] SIDN Labs. 2019. .nl stats and data. https://stats.sidnlabs.nl/en/dns.html.

[63] SWITCH. 2019. DNS Queries: UDP Compared to TCP. https://www.nic.ch/statistics/dns/udp-tcp/.

[64] University of Oregon. 2019. Route Views Project. http://www.routeviews.org/routeviews/.

[65] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. 2016. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1877–1888.

[66] Stefan Viehböck. 2018. Millions of Xiongmai Video Surveillance Devices Can be Hacked via Cloud Feature (XMEye P2P Cloud). https://sec-consult.com/en/blog/2018/10/millions-of-xiongmai-video-surveillance-devices-can-be-hacked-\via-cloud-feature-xmeye-p2p-cloud/.

[67] Paul Vixie. 2007. DNS complexity. *Queue* 5, 3 (2007), 24–29.

[68] Duane Wessels. 2018. ipv4-heatmap tool. https://github.com/measurement-factory/ipv4-heatmap.

[69] D. Wing and A. Yourtchenko. 2012. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555 (Proposed Standard). , 15 pages. https://doi.org/10.17487/RFC6555 Obsoleted by RFC 8305.

[70] Maarten Wullink, Giovane CM Moura, Moritz Müller, and Cristian Hesselman. 2016. ENTRADA: A high-performance network traffic data streaming warehouse. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 913–918.