Recursives in the Wild: Engineering Authoritative DNS Servers

Moritz Müller
SIDN Labs and University of Twente

Giovane C. M. Moura
SIDN Labs

Ricardo de O. Schmidt
SIDN Labs and University of Twente

John Heidemann
USC/Information Sciences Institute

ABSTRACT

In Internet Domain Name System (DNS), services operate *authoritative* name servers that individuals query through *recursive resolvers*. Operators strive to provide reliability by operating multiple name servers (NS), each on a separate IP address, and by using IP anycast to allow NSes to provide service from many physical locations. To meet their goals of minimizing latency and balancing load across NSes and anycast, operators need to know how recursive resolvers select an NS, and how that interacts with their NS deployments. Prior work has shown some recursives search for low latency, while others pick an NS at random or round robin, but did not examine how prevalent each choice was. This paper provides the first analysis of how recursives select between name servers in the wild, and from that we provide guidance to operators how to engineer their name servers to reach their goals. We conclude that all NSes need to be equally strong and therefore we recommend to deploy IP anycast at every single authoritative.

CCS CONCEPTS

• **Networks → Network design principles**; **Network measurement**; **Naming and addressing**; *Network layer protocols*; *Network resources allocation*; *Network performance analysis*; *Denial-of-service attacks*; *Logical / virtual topologies*; *Overlay and other logical network structures*;

KEYWORDS

DNS, recursive DNS servers, authoritative DNS servers, anycast

ACM Reference Format:
Moritz Müller, Giovane C. M. Moura, Ricardo de O. Schmidt, and John Heidemann. 2017. Recursives in the Wild: Engineering Authoritative DNS Servers . In *Proceedings of IMC '17, London, United Kingdom, November 1–3, 2017,* 7 pages.
https://doi.org/10.1145/3131365.3131366

1 INTRODUCTION

The Internet Domain Name System (DNS) puts the "dot" in .com, providing a global naming service for web, e-mail and all Internet services [16]. DNS is a distributed system with a hierarchical namespace where each component (the root, .org and wikipedia.org) is served by *authoritative servers*. For each component, NS (name server) records specify the hosts that act as authoritative servers [17]. To use the DNS, a user's browser or operating system employs a *stub resolver* to place a query. It then talks to a *recursive resolver* that walks through authoritative servers for each level of the DNS hierarchy, possibly using prior cached results.

DNS operators face numerous challenges when engineering their services, including providing fault tolerance, increasing the resilience against denial-of-service (DoS) attacks, and reducing latency. In this paper, we focus on latency. DNS can be a noticeable part of web latency [28], so users, web browser authors, and DNS service providers strive to reduce latency through DNS server replication [17] and IP anycast [15, 21].

Today most large DNS services replicate hosts specified in NS records to many physical *sites* with *IP anycast*. Sites that belong to one NS record form an *anycast service*. Important DNS services such as the DNS Root are very widely replicated, with 13 different anycast services (each a *root letter*), each with a distinct IP address in distinct ASes [12]. Each letter has multiple sites, with 500 across all letters [24]. These practices are common in all important domains. All top-level domains (TLDs) run at least two different authoritatives on distinct IP addresses. For example the Netherlands, .nl, has 8 separate authoritatives, of which 5 are unicast and 3 are anycast services deployed across more than 80 sites.

A DNS operator is faced with a challenge: how many authoritatives should they operate? How many should be anycast services, and how many sites should each anycast service employ? Each authoritative and site brings cost and some complexity. Recent work has suggested that a few IP anycast sites can provide good latency for a global DNS service [25], but what happens to overall performance of a DNS service that is composed of different authoritative nameservers, some of which are anycast services and some of which may be unicast?

Answering these questions when engineering a DNS service is challenging because little is known about the recursive resolvers that make requests. There are many different implementations of recursive resolvers with a multitude of software releases, how they select between authoritative servers is not defined, and we cannot determine which implementations run where, nor how many of each exist. Early work [33] shows that the behavior across different recursive resolvers is diverse, with some making intentional choices

IMC '17, November 1–3, 2017, London, United Kingdom

ACM ISBN 978-1-4503-5118-8/17/11…$15.00
https://doi.org/10.1145/3131365.3131366

and others alternating across all NSes for a service. While this result has been reconfirmed, to our knowledge, there is no public study on how this interacts with different design choices of name server deployments, nor how it should influence its design.

The first contribution of this paper is to *re-evaluate how recursive resolvers select authoritative name servers* (§4), but in the wild, with the goal of learning from the *aggregate* behavior in order to better engineer authoritative deployments. We answer this question with a controlled study of an experimental, worldwide, name server deployment using Amazon Web Services (AWS) coupled with global data from the Root DNS servers and the .nl TLD (§5). Our key results are that most recursives check all authoritatives over time (§4.1), about half of recursives show a preference based on latency (§4.2), and that these preferences are most significant when authoritatives have large differences in latency (§4.3).

Based on these findings, our second contribution is to suggest *how DNS operators can optimize a DNS service* to reduce latency for diverse clients (§7). In order to achieve optimal performance we conclude that all NSes need to be equally strong and therefore recommend to use anycast at all of them. This new recommendation augments existing practices about operation of individual anycast services [1, 15], with advice about DNS services that employ multiple NSes.

2 BACKGROUND: OPERATING DNS

Figure 1 shows the relationship between the main elements involved in the DNS ecosystem. Each *authoritative server* (AT) is identified by a domain name, stored in an NS record, which can be reachable by one or multiple IP addresses. Operators often mix unicast and anycast services across their authoritatives, and there is no consensus on how many NSes is the best. For example, most of TLDs within the root zone use 4 NSes, but some use up to 13, and each of these NSes can be replicated and globally distributed using IP anycast and load balancers [18].Second level domains like example.com under TLDs like .com, .net and .org have a median of 2 NS records (mean of 2.3, 2.4, and 2.4n) and the domain names of .nl have a median of 3 NS records (mean of 2.6 as of 2017-08-01).

Recursive resolvers (*R* in Figure 1) answer to DNS queries originated at clients (*CL* in Figure 1) by either finding it in their local cache, or sending queries to authoritative servers to obtain the final answer to be returned to the client [10]. Besides the local cache with information on DNS records, many recursives also keep an *infrastructure cache* with information on the latency (Round Trip Time, RTT) of each queried authoritative server, grouped by IP address. The infrastructure cache is used to make informed choices among multiple authoritatives for a given zone. For example, Unbound [30] implements a smoothed RTT (SRTT), and BIND [3] an SRTT with a decaying factor. Some implementations of recursive resolvers, particularly those for embedded devices like home routers, may omit the infrastructure cache.

3 MEASUREMENTS AND DATASETS

Next we describe how we measure the way recursives choose authoritative servers, using both active measurements and passive observations of production DNS at the root and .nl. Our work focuses on measurements from the field, so that we capture the actual range of current behavior, and to evaluate *all* currently used recursives. (Our work therefore complements prior studies that examine specific implementations in testbeds [33]. Their work are definite about why *a* recursive makes a choice, but not on *how many* such recursives are in use.)

ID	locations (airport code)	VPs
2A	GRU (São Paulo, BR), NRT (Tokyo, JP)	8,702
2B	DUB (Dublin, IE), FRA (Frankfurt, DE)	8,685
2C	FRA, SYD (Sydney, AU)	8,658
3A	GRU, NRT, SYD	8,684
3B	DUB, FRA, IAD (Washington, US)	8,693
4A	GRU, NRT, SYD, DUB	8,702
4B	DUB, FRA, IAD, SFO (San Francisco, US)	8,689

Table 1: Combinations of authoritatives we deploy and the number of VPs they see.

3.1 Measurement Design

To observe recursive-to-authoritative mapping on the Internet, we deploy authoritative servers for a test domain (ourtestdomain.nl) in 7 different datacenters, all reachable by a distinct IPv4 unicast address. Sites are hosted by Amazon, using NSD 4.1.7 running on Ubuntu Linux on AWS EC2 virtual machines.

We then resolve names serviced by this test domain from about 9,700 vantage points (VPs) distributed over 3,300 Autonomous Systems (ASes) (of which 1,040 ASes host 2 or more probes), all the RIPE Atlas probes that are active when we take each measurement [23]. Each VP is a DNS client (a CL in Figure 1) that queries for a DNS TXT resource record using an IPv4 address.

Each VP uses whatever their local configured recursive is. Those recursives are determined by the individual or ISP hosting each VP. Overall, we observe over 11,000 unique IP addresses of upstream recursives at our authoritatives, located in over 2,500 ASes.

To determine which authoritative NS the VP reaches, we configure each NS with a *different* response for the same DNS TXT resource. While most studies of anycast catchment use DNS CHAOS-class queries, where a query on the `hostname.bind` or `id.server` identifies a specific authoritative [31], CHAOS queries would be answered directly by the configured recursive server. We use Internet-class queries that pass through a recursive to the authoritative. The resulting dataset from the processing described is publicly available at our website [19] and at RIPE Atlas [22].

Cold caches. DNS responses are extensively cached [6]. We insure that caches do not interfere with our measurements in several ways: our authoritatives are used only for our test domain, we set the time-to-live (TTL) [16] of the TXT record to 5 seconds, use unique labels for each query, and run separate measurements with a break of at least 4 hours, giving recursives ample time to drop the IP addresses of the authoritatives from their infrastructure caches.

Authoritatives location. We deploy 7 combinations of authoritative servers located around the globe (Table 1). We identify each by the number of sites (2 to 4) and a variation (A, B, or C). The combinations vary geographic proximity, with the authoritatives close to each other (2B, 3B, 4B) or farther apart (2A, 2C, 3A, 4A). For each combination we determine the recursive-to-authoritative mapping with RIPE Atlas, querying the TXT record of the domain name every 2 minutes for 1 hour. We choose 2 to 4 name servers

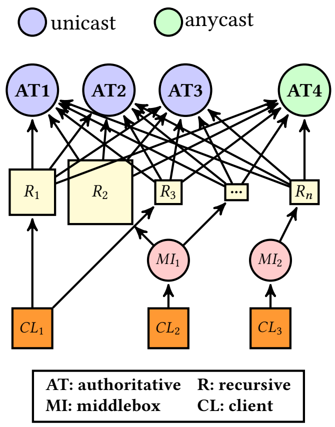

Figure 1: TLD Setup, Recursives, Middleboxes and Clients.

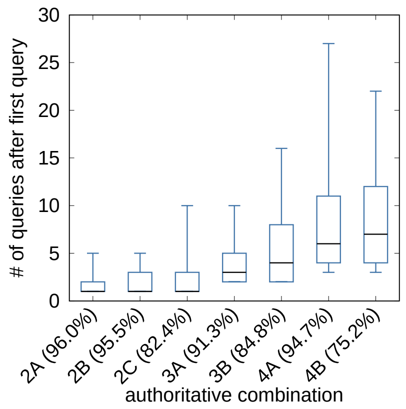

Figure 2: Queries to probe all authoritatives, after the first query. (Boxes show quartiles and whiskers 10/90%ile.)

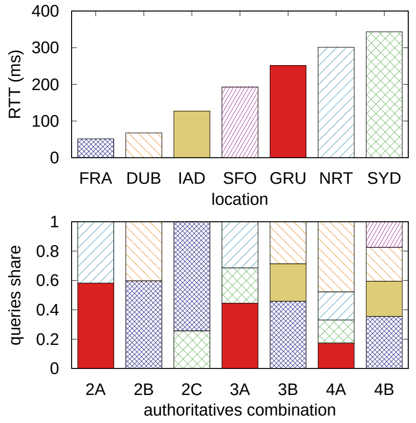

Figure 3: Median RTT (top) and query distribution (bottom) for combinations of authoritatives.

because it reflects the most common name server deployments and is enough to provide geographic diversity. While we consider "only" one hour of data, it seems unlikely that authoritative selection is strongly affected by diurnal factors.

Measurement challenges and considerations. We consider several challenges that might interfere with our measurements.

Atlas probes might be configured to use multiple recursives and, therefore, in our analysis we consider unique combinations of probe ID and recursive IP as a single VP (or client, in Figure 1);

Middleboxes (load balancers, DNS forwarders) between VPs and recursives (MI in Figure 1) or recursives which use anycast may interfere, causing queries to go to different recursives or to warm up a cache. Full studies of DNS resolution are quite involved [26] and outside the scope of this paper. We confirm that middleboxes have only minor effects on our data by comparing client and authoritative data. Specifically, we compare Figure 4 to the same plot using data collected at the authoritatives for all recursives that send at least five queries during one measurement (graph omitted due to space). The two graphs are basically equivalent, suggesting that middleboxes do not significantly distort what we see at the clients.

Because of the use of these middleboxes we refrain from trying to identify the implementations of the recursives directly.

Our VPs (RIPE Atlas probes) are unevenly distributed around the globe, with far more in Europe than elsewhere [4, 5, 25]. To take this uneven distribution into account when we study geographic effects, we group probes by continent and analyze them individually in most research questions.

We focus on UDP DNS for IPv4, not TCP or IPv6. The majority of our VPs have IPv4 connectivity only [4] (69%) and so fully study of IPv6 does not make sense. However, we verify that our results apply to IPv6 by repeating a subset of our measurements there. We use the VPs capable of IPv6 to query authoritatives reachable only via IPv6 addresses and we confirm that, overall, recursives follow the same strategy when querying via IPv6 (graph omitted due to space, but available at [20]). We focus on DNS over UDP because it is by far the dominant transport protocol today (more than 97% of connections for .nl [27] and most Root DNS servers [11]).

Finally, our results are based on one service, the country-code (ccTLD) for the Netherlands (.nl). Our results are about recursive and authoritative resolvers and are not specific to this domain. We believe our results generalize to other domains (both ccTLDs and general TLDs), but additional study is needed.

3.2 Root DNS and TLD data

We use passive measurements from the DITL (Day In The Life of the Internet) [8], collected on 2017-04-12 at 10 Root DNS letters (B, G and L are missing). We look at the one-hour sample from 12:00 to 13:00 (UTC), since that duration is sufficient to evaluate our claims. By default, most implementations of recursive resolvers do not treat Root DNS servers different from other authoritatives.

We also use traffic collected at 4 authoritative servers of the .nl ccTLD [32]. For consistency, we use .nl traces from the same time slot as of DITL data. We use these data sets to validate our observations from §3.1. Note that we cannot enforce a *cold cache* condition in these passive measurements such that a recursive could already prefer an authoritative, and RTT data is not available.

4 ANALYSIS OF RECURSIVE BEHAVIOR

4.1 Do recursives query all authoritatives?

Our first question is to understand how many recursive resolvers query *all* available authoritative servers. Figure 2 shows how many queries, after the very first one, it takes for a recursive to probe all available authoritatives (2 to 4 depending on the configuration from Table 1).

The percentage of recursives that query all available authoritatives is given in the x-axis labels of Figure 2. Most recursives query all authoritatives (75 to 96%), and with two authoritatives (2A, 2B, 2C) half the recursives probe the second authoritative already on their second query; but with four authoritatives (4A, 4B) it takes a median of up to 7 queries for the recursives to query them all. Operators can conclude that all their authoritatives are visible to most recursives.

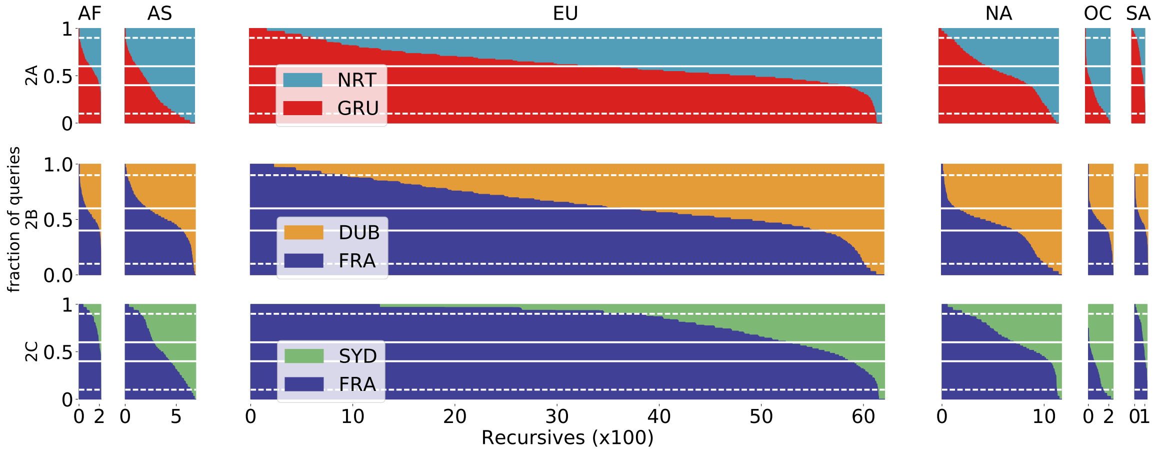

Figure 4: Recursive queries distribution for authoritative combinations 2A (top), 2B (center) and 2C (bottom). Solid and dotted horizontal lines mark VPs with weak and strong preference towards an authoritative.

4.2 How are queries distributed per authoritative over time?

Since most recursives query all available authoritative servers relatively quickly, we next look at how queries are spread over multiple authoritatives, and if this is affected by RTT. Here, our analysis starts once each recursive reaches a hot-cache condition by querying all authoritatives at least once.

Figure 3 compares the fraction of queries (bottom) received by each authoritative with the median RTT (top) from the recursives to that authoritative. We see that authoritatives with lower RTTs are often favored; *e.g.,* FRA has the lowest latency (51 ms) and always sees most queries overall.

When running multiple authoritative servers, the operator should expect an uneven distribution of queries among them. Servers to which clients see shorter RTT will likely receive most queries.

Our findings in this section, and in §4.1, confirm those of previous work by Yu *et al.* [33], in which authors show that 3 out of 6 recursive implementations are strongly based on RTT. However, unlike the previous work, our conclusions are drawn from real-world observations instead of experimental setup and predictions based on algorithms.

4.3 How do recursives distribute queries?

We now look at how individual recursives in the wild distribute their queries across multiple options of authoritatives.

Figure 4 shows the individual preferences of recursives (VP/recursive pair, grouped by continent) when having the choice between two authoritatives. The x-axis of Figure 4 displays all recursives, and the y-axis gives the fraction of queries every recursive sends to each authoritative. Table 2 summarizes these results.

In order to quantify *how many* recursives are actually RTT based, we consider only VPs that experience a difference in median RTT of at least 50 ms between the authoritatives[1]. Based on our observations we define two thresholds for recursive preference: a *weak* preference if the recursive sends at least 60% of its queries to one authoritative (solid lines in Figure 4), and a *strong* preference if at least 90% of queries go to one authoritative (dotted lines in Figure 4).

config:	2A				2B				2C			
cont-	NRT		GRU		FRA		DUB		FRA		SYD	
ient	%	RTT	%	RTT	%	RTT	%	RTT	%	RTT	%	RTT
AF	39	467	61	393	57	200	43	204	85	200	15	513
AS	70	130	30	353	53	241	47	261	54	200	46	193
EU	37	310	63	248	65	39	35	53	83	39	17	355
NA	46	190	54	173	41	162	59	152	66	149	33	237
OC	74	201	26	363	46	346	54	335	22	370	78	48
SA	27	364	73	102	49	259	51	259	70	258	30	399

(AF: Africa, AS:Asia, EU: Europe, NA: North America,
OC: Oceania, SA: South America)

Table 2: Query distribution and median RTT (ms) for VPs grouped by continent and three different combinations of authoritatives (Table 1).

We see that 61% of recursives in 2A (top), 59% in 2B (center) and 69% in 2C have at least a weak preference; and 10%, 12% and 37% have a strong preference in 2A, 2B, and 2C respectively. After sending queries for 30 minutes, recursives with a weak preference develop an even stronger preference (omitted due to space, but available at [20]).

The distribution of queries per authoritative is inversely proportional to the median RTT to each recursive. The bottom plot of Figure 4 clearly shows this point, where there is a strong bias for VPs in Europe (EU): VPs largely prefer FRA (Frankfurt) over SYD (Sydney); and the opposite for VPs in Oceania (OC): SYD over FRA.

By contrast, when given a choice between two roughly equidistant authoritatives, there is a more even split. We see a roughly even split both when the recursives are near, with Europe going to Frankfurt and Dublin (configuration 2B, EU to FRA and DUB), or far, where they go to Brazil and Japan (configuration 2A, EU to GRU and NRT). Some VPs still have a preference; we assume

[1] We think that it is reasonable for a recursive to prefer an authoritative over another when it responds at least 50 ms faster.

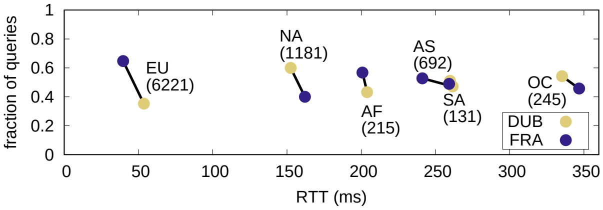

Figure 5: RTT sensitivity of 2B (number of VPs in brackets)

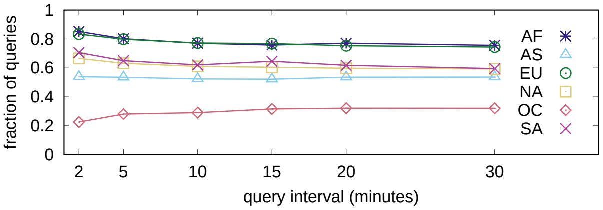

Figure 6: Fraction of queries to FRA (remainder go to SYD, configuration 2C), as query interval varies from 2 to 30 minutes.

these represent VPs in Ireland or Germany. Thus, DNS operators can expect that the majority of recursives will send most queries to the fastest responding authoritative. However, a significant share of recursives (in case of 2B up to 41%) also send up to 40% of their queries to the slower responding authoritative.

To expand on this result, Figure 5 compares the median RTT between VPs that go to a given site and the fraction of queries they send to that site, again grouped by continent. Differences between the two points for each continent indicate a spread in preference (differences in queries on the y axis) or RTT (differences in the x axis). We show the results for 2B because in this setup, both authoritatives are located rather close to each other such that the VPs should see a similar RTT for both of them. We see that recursives in Europe that prefer Frankfurt do so because of lower latency (EU VPs that prefer FRA have 13.9 ms lower latency than DUB). In contrast, recursives in Asia distribute queries nearly equally, in spite of a similar difference in latency (AS VPs see 20.3 ms difference). We conclude that *preferences based on RTT decrease when authoritatives are far away* (when they have large median RTT, roughly more than 150 ms). As a consequence, DNS operators who operate two authoritatives close to each other can expect a roughly equal distribution from recursives further away and a preference from recursives closer by.

4.4 How does query frequency influence selection?

Many recursive resolvers track the latency to authoritatives (§2), but how long they keep this information varies. By default, BIND [3] caches latency for 10 minutes, and Unbound caches it for about 15 minutes [30]. In this section, we measure the influence of frequency of queries in the selection of authoritatives by the recursives. To do that, we repeat the measurement for configuration 2C. However, instead of a 2-minute interval between queries, we probe every 5, 10, 15, and 30 minutes. We choose 2C because, in this setup, we observe the strongest preference for one of the two recursives.

We show these results in Figure 6. We see that *preferences for authoritatives are stronger when probing is very frequent, but persist with less frequent queries*, particularly at 2 minute intervals. Beyond 10 minutes, the preferences are fairly stable, but surprisingly continue. This result suggests that recursive preference often persist beyond the nominal 10 or 15 minute timeout in BIND and Unbound and therefore, also recursives that query only occasionally the name servers of an operator can still benefit from a once learned preference.

5 RECURSIVE BEHAVIOR TOWARDS AUTHORITATIVES IN PRODUCTION

After analyzing behavior of the recursive resolver for each RIPE Atlas VP in our measurement (§4), we now focus on validating the results by looking at DNS traffic of production deployments of the Root DNS zone and the .nl ccTLD.

Root: We use DITL-2017 [8] traffic from 10 out of 13 Root letters (B, G and L were missing at the point of our analysis) to analyze queries to the root servers (root *letters*). Figure 7 (top) shows the distribution of queries of recursives that sent at least 250 queries to the root servers in one hour. For each VP, the top color band represents the letter it queries most, with the next band its second preferred letter, etc.

While we find that almost all recursives tend to explore all authoritatives (§4.1), many recursives (about 20%) send queries to only one letter. The remainder tend to query many letters (60% query at least 6), but only 2% query all 10 authoritatives. One reason this analysis of Root traffic differs from our experiment is that here we cannot "clear" the client caches, and most recursives have prior queries to root letters.

The .nl ccTLD: the picture slightly changes for queries to a ccTLD. In the bottom plot of Figure 7 we plot the distribution of .nl authoritatives. The majority of recursives query all the authoritatives which confirms our observations from our test deployment. Here, the number of recursives that query only authoritatives is also smaller than at the Root servers.

We conclude that recursive behavior at the Root and at a TLD is comparable with our testbed, except that a much larger fraction of resolvers have a strong preference for a particular Root letter. The majority of the recursives send queries to every available authoritative.

6 RELATED WORK

To the best of our knowledge, this is the first extensive study that investigates how authoritative server load is affected by the choices recursives resolvers make.

The study by Yu *et al.* [33] considers the closely related question of how different recursives choose authoritatives. Their approach is to evaluate different implementations of recursive resolvers in a controlled environment, and they find that half of the implementations choose the authority with lowest latency, while the others choose randomly (although perhaps biased by latency). Our study complements theirs by looking at what happens in practice, in effect weighing their findings by the diverse set of software and latencies

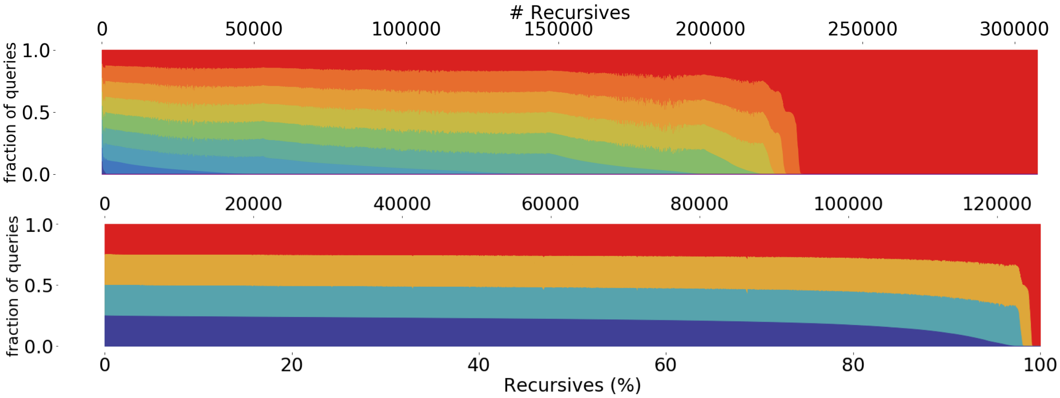

Figure 7: Distribution of queries of recursives with at least 250 queries across 10 out of 13 Root letters (top) and across 4 out of 8 name servers of .nl (bottom).

seen across the 9,000 vantage points, and by all users of the Root DNS servers and .nl ccTLD.

Kührer *et al.* [14] evaluates millions of general open recursives resolvers. They consider open recursive response authenticity and integrity, distribution of device types, and their potential role in DNS attacks. Although similar to our work, they focus on external identification and attacks, not "regular" recursive use. (Using open recursive resolvers in our study for additional measurements is possible future work.)

Also close to our work, Ager *et al.* [2] examine recursive resolution at 50 ISPs and Google Public DNS and OpenDNS. Our study considers many more recursives (more than 9,000 locations in RIPE Atlas), and we focus on the role those recursives have in designing an authoritative server system.

Schomp *et al.* [26] consider the client-side of recursive resolvers. Unlike our work, they do not discuss implications for DNS operators. In another work, Korczyński *et al.* [13] have identified second-level domains in the wild whose authoritative DNS servers vulnerable to zone poisoning through dynamic DNS updates [29]. While their work analyzes authoritative servers, it focus on the management of zone files, while we focus on how recursives choose authoritatives.

Finally, other studies such as Castro *et al.* [7] have examined DNS traffic at the Root DNS servers. They often use DITL data (as we do), but typical focus on client performance and balance of traffic across the Root DNS servers, rather than the design of a specific server infrastructure.

7 RECOMMENDATIONS AND CONCLUSIONS

Our main contribution is the analysis of how recursives choose authoritatives in the wild, and how that can influence the design of authoritative server systems. We present the following recommendations for DNS providers:

Primary recommendation: when optimizing user latency, *worst-case latency will be limited by the least anycast authoritative.* The implication is that if some authoritatives in a server system are anycast, *all* should be. We have shown that most recursives will always send some queries to all authoritatives of a service. Even if one or some authoritatives employ large anycast networks for low latency, recursives will still send some queries to the remaining unicast sites, which implies higher latency. These unicast sites might respond with a short RTT to some clients nearby, but not to clients that are further away and that could be served by other (anycast) sites faster. Overall improvement in latency depends on the distribution of clients and also their caching management policy; possible future work is to model or measure that improvement.

While it may seem obvious that all authoritatives should be equal capacity, the importance this relationship is not always clear when making deployment decisions. A DNS operator may seek to improve latency by adding an additional authoritative provided by a large, third-party DNS provider to their current operations, yet not get full value if the two authoritative have different capacity.

SIDN operates .nl, and for us this principle suggests adjusting our architecture. We currently have 5 unicast authoritatives in the Netherlands, and three authoritatives that are anycast with sites around the world. Although the anycast authoritatives can offer lower latency to users from North America, 23% of incoming queries to the unicast name servers in the Netherlands are from the U.S. [27], experiencing worse latency than they might otherwise. Examine of other TLD services is potential future work.

Other Considerations: Other reasons motivate multiple authoritatives per service, or large use of anycast. Anycast is important to mitigate DDoS attacks [18]. In addition, standard practices recommend multiple authoritatives in different locations for fault tolerance [9]. DNS operators should also be aware of the deployment complexity that anycast might incur when compared to unicast [15].

For latency, prior work has shown that relatively few well-peered anycast sites, well-connected with the important clients, can provide good global latency [25]. We add to this advice on that all authoritatives have to provide low latency to reduce overall service latency to users of most recursives.

Conclusion: In this paper we have shown the diverse server selection strategies of recursives in the wild. While many select authoritatives preferentially to reduce latency, some queries usually go to all authoritatives. The main implication of these findings is that all name servers in a DNS service for a zone need to be consistently provisioned (with reasonable anycast) to provide consistent low latency to users.

Acknowledgments

We would like to thank Marco Davids and Marc Groeneweg for their support in this research. This research has been partially supported by measurements obtained from RIPE Atlas, an open measurements platform operated by RIPE NCC, as well as by the DITL measurement data made available by DNS-OARC.

Moritz Müller, Giovane C. M. Moura, and Ricardo de O. Schmidt developed this work as part of the SAND project (http://www.sand-project.nl).

John Heidemann's work in this paper is partially sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, via BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate (agreement FA8750-12-2-0344) and via contract number HHSP233201600010C. The U.S. Government is authorized to make reprints for governmental purposes notwithstanding any copyright. The views contained herein are those of the authors and do not necessarily represent those of NSF, DHS or the U.S. Government.

REFERENCES

[1] Abley, J., and Lindqvist, K. Operation of Anycast Services. RFC 4786 (Best Current Practice), Dec. 2006.

[2] Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S. Comparing dns resolvers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet Measurement* (Sept. 2010), ACM, pp. 15–21.

[3] Almond, C. Address database dump (ADB) - understanding the fields and what they represent. https://kb.isc.org/article/AA-01463/0/Address-database-dump-ADB-understanding-the-fields-and-what-they-represent.html, 2017.

[4] Bajpai, V., , Eravuchira, S., Schönwälder, J., Kisteleki, R., and Aben, E. Vantage Point Selection for IPv6 Measurements: Benefits and Limitations of RIPE Atlas Tags. In *IFIP/IEEE International Symposium on Integrated Network Management (IM 2017)* (Lisbon, Portugal, May 2017).

[5] Bajpai, V., Eravuchira, S. J., and Schönwälder, J. Lessons learned from using the RIPE Atlas platform for measurement research. *SIGCOMM Comput. Commun. Rev. 45*, 3 (July 2015), 35–42.

[6] Callahan, T., Allman, M., and Rabinovich, M. On modern DNS behavior and properties. *ACM SIGCOMM Computer Communication Review 43*, 3 (July 2013), 7–15.

[7] Castro, S., Wessels, D., Fomenkov, M., and Claffy, K. A Day at the Root of the Internet. *ACM Computer Communication Review 38*, 5 (Apr. 2008), 41–46.

[8] DNS OARC. DITL Traces and Analysis. https://www.dns-oarc.net/oarc/data/ditl/2017, Feb. 2017.

[9] Elz, R., Bush, R., Bradner, S., and Patton, M. Selection and Operation of Secondary DNS Servers. RFC 2182 (Best Current Practice), July 1997.

[10] Hoffman, P., Sullivan, A., and Fujiwara, K. DNS Terminology. RFC 7719 (Informational), Dec. 2015.

[11] ICANN. RSSAC002: RSSAC Advisory on Measurements of the Root Server System. https://www.icann.org/en/system/files/files/rssac-002-measurements-root-20nov14-en.pdf, Nov. 2014.

[12] Internet Assigned Numbers Authority (IANA). Technical requirements for authoritative name servers. https://www.iana.org/help/nameserver-requirements, 2017.

[13] Korczyński, M., Król, M., and van Eeten, M. Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates. In *Proceedings of the 2016 ACM on Internet Measurement Conference* (2016), ACM, pp. 271–278.

[14] Kührer, M., Hupperich, T., Bushart, J., Rossow, C., and Holz, T. Going wild: Large-scale classification of open DNS resolvers. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference* (Oct. 2015), ACM, pp. 355–368.

[15] McPherson, D., Oran, D., Thaler, D., and Osterweil, E. Architectural Considerations of IP Anycast. RFC 7094 (Informational), Jan. 2014.

[16] Mockapetris, P. Domain names - concepts and facilities. RFC 1034, Nov. 1987.

[17] Mockapetris, P. Domain names - implementation and specification. RFC 1035, Nov. 1987.

[18] Moura, G. C. M., de O. Schmidt, R., Heidemann, J., de Vries, W. B., Müller, M., Wei, L., and Hesselman, C. Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. In *Proceedings of the 2016 ACM Conference on Internet Measurement Conference* (Oct. 2016), pp. 255–270.

[19] Müller, M., Moura, G. C. M., de O. Schmidt, R., and Heidemann, J. Recursives in the wild datasets. https://www.simpleweb.org/wiki/index.php/Traces#Recursives_in_the_Wild:_Engineering_Authoritative_DNS_Servers and https://ant.isi.edu/datasets/all.html#DNS_Recursive_Study-20170323, May 2017.

[20] Müller, M., Moura, G. C. M., de O. Schmidt, R., and Heidemann, J. Recursives in the Wild: Engineering Authoritative DNS Servers. Tech. Rep. ISI-TR-720, USC/Information Sciences Institute, Sept. 2017. http://www.isi.edu/%7ejohnh/PAPERS/Mueller17a.html.

[21] Partridge, C., Mendez, T., and Milliken, W. Host Anycasting Service. RFC 1546 (Informational), Nov. 1993.

[22] RIPE NCC. RIPE Atlas measurement ids. https://atlas.ripe.net/measurements/ID, Mar. 2017. ID is the experiment ID: 2A: 7951948, 2B: 7953390, 2C: 7967380, 3A: 7961003, 3B: 7954122, 4A: 7966930, 4B: 7960323, 2C-5min: 8321846, 2C-10min: 8323303, 2C-15min: 8324963, 2C-20min: 8329423, 2C-15min: 8335072.

[23] RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal (IPJ) 18*, 3 (Sep 2015), 2–26.

[24] Root Server Operators. Root DNS, Feb. 2017. http://root-servers.org/.

[25] Schmidt, R. d. O., Heidemann, J., and Kuipers, J. H. Anycast latency: How many sites are enough? In *Proceedings of the Passive and Active Measurement Workshop* (Sydney, Australia, Mar. 2017), Springer, pp. 188–200.

[26] Schomp, K., Callahan, T., Rabinovich, M., and Allman, M. On measuring the client-side DNS infrastructure. In *Proceedings of the* (Barcelona, Spain, Oct. 2013).

[27] SIDN Labs. .nl stats and data, Mar. 2017. http://stats.sidnlabs.nl/#network.

[28] Singla, A., Chandrasekaran, B., Godfrey, P., and Maggs, B. The internet at the speed of light. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (Oct. 2014), ACM, pp. 1–7.

[29] Vixie, P., Thomson, S., Rekhter, Y., and Bound, J. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard), Apr. 1997. Updated by RFCs 3007, 4035, 4033, 4034.

[30] Wijngaards, W. Unbound Timeout Information. https://unbound.net/documentation/info_timeout.html, Nov. 2010.

[31] Woolf, S., and Conrad, D. Requirements for a mechanism identifying a name server instance. RFC 4892, Internet Request For Comments, June 2007.

[32] Wullink, M., Moura, G. C., Müller, M., and Hesselman, C. Entrada: A high-performance network traffic data streaming warehouse. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP* (Apr. 2016), IEEE, pp. 913–918.

[33] Yu, Y., Wessels, D., Larson, M., and Zhang, L. Authority Server Selection in DNS Caching Resolvers. *SIGCOMM Computer Communication Review 42*, 2 (Mar. 2012), 80–86.