Recursives in the Wild: Engineering Authoritative DNS Servers (extended)

ISI-TR-720 1 June 2017

Moritz Müller[1,2] Giovane C. M. Moura[1]
Ricardo de O. Schmidt[1,2] John Heidemann[3]
1: SIDN Labs 2: University of Twente 3: 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 name server operators 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.

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 [14]. DNS is a distributed service with a hierarchical namespace where each component (the root, `.org` and `wikipedia.org`) is served by *authoritative servers*. Often multiple authoritative servers provide the same component with different goals including reducing latency, providing fault tolerance, and to help mitigate denial-of-service (DoS) attacks. 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 the DNS hierarchy, possibly using prior cached results.

DNS can be a noticeable part of web latency [25], so users, web browser authors, and DNS service providers strive to reduce latency through DNS server replication [15] and IP anycast [18, 13]. DNS authoritative servers can be replicated, with multiple servers identified to recursive resolvers through *NS* records [15], each pointing at one or multiple IP addresses (*e.g.,* one IPv4 and another IPv6 address).

Today most large DNS services replicate name servers to many physical locations with *IP anycast*. Important services such as the DNS Root are very widely replicated, with 13 different name servers (each a *root letter*), all with multiple sites, totaling over 500 anycast sites [21] with distinct IP addresses in distinct ASes [11]. Also top-level domains (TLDs) run at least two different authoritatives with two distinct IP addresses; for example, `.nl` (the Netherlands) has 8 separate servers, of which 5 are unicast and 3 use anycast across more than 80 sites.

A DNS operator is faced with a challenge: how many servers should they operate? How many should use anycast, and how many sites should each anycast service employ? Recent work has suggested few IP anycast instances can provide good global latency [22] for one name server, but is one anycast service enough to improve latency?

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, and how they select between authoritative servers is not defined. Early work [29] shows that the behavior across different recursive resolvers is diverse, with some making intentional choices 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 aggregated 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 authoritives over time (§4.1),

about half of recurisves show a preference based on latency (§4.2), and that these preferences are most signficant when authoratives have large differences in latency (§4.3).

Based on thse 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 every single one of them.

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 strategies 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 [16][1] .

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 [9]. 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 [26] implements a smoothed RTT (SRTT), and BIND [2] 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 [29]. Their work are definite about why *a* recursive makes a choice, but not on *how many* such recursives are in use.)

3.1 Measurement Design

[1]Figure 9 shows the number of NS records of TLDs in the root zone.

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.

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), all the RIPE Atlas probes that are active when we take each measurement [20]. Each VP is a DNS client (a CL in Figure 1) that queries for a DNS TXT resource record over IPv4. (In principle we could also query over IPv6; we focus on IPv4 for now because 69% of VPs lack IPv6 support.) Each VP uses whatever their local configured recursive is. Those recursives are determined by the individual or ISP hosting each VP.

To determine which authoritative NS the VP reaches, we configure each with a *different* response for the same DNS TXT resource. We choose TXT records because a DNS CHAOS query [27] would be responded by the recursive and not by the authoritative. The resulting dataset from the processing described is publicly available at our website [17] and at Ripe Atlas [19].

Cold caches. DNS responses are extensively cached [5]. 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) [14] 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.

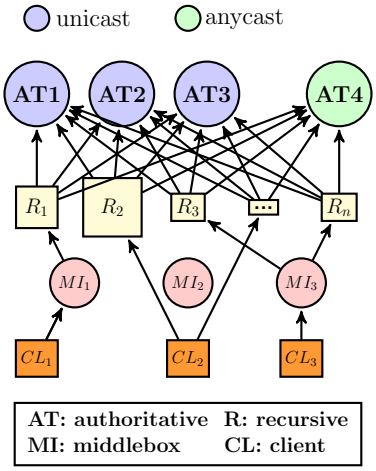

Figure 1: TLD Setup, Recursives, Middleboxes and Clients.

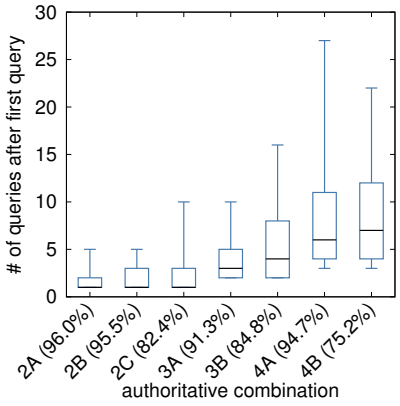

Figure 2: Queries to probe all authoritatives, after the first query. (Boxes show quartiles and whiskers 10/90%ile.)

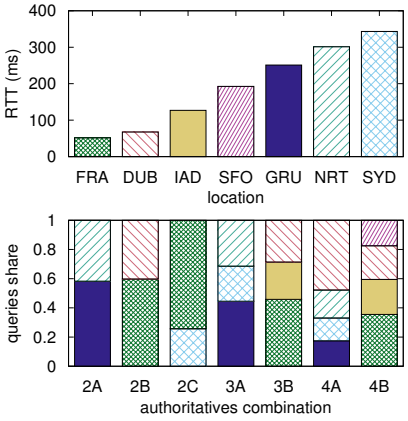

Figure 3: Query distribution (top) and median RTT (bottom) for combinations of authoritatives.

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

Atlas probes might be configured via DHCP 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. We cannot eliminate their effects, but we confirm that they 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 (see Figure 8).

The two graphs are basically equivalent, suggesting that middleboxes do not significantly distort what we see at the clients.

To take RIPE Atlas' large number of probes in Europe [4, 22, 3] into account, we group probes by continent and analyze them individually in most research questions.

We focus on UDP DNS for IPv4 only, not TCP or IPv6. IPv6 is future work; we cannot study IPv6 at this time because the majority of our VPs only have IPv4 connectivity [3]. We focus on DNS over UDP because it is by far the dominant transport protocol today (more than 97% of connections for `.nl` and most DNS roots).

3.2 Root DNS and TLD data

We use passive measurements from the DITL (Day In The Life of the Internet) [7], collected on 2017-04-12 at 10 Root DNS servers (B, G and L are missing). We look at the one-hour sample from 12:00 to 13:00 (UTC), since that duration 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 ccTLD `.nl` [28]. For consistency, we use captures 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, 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.

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

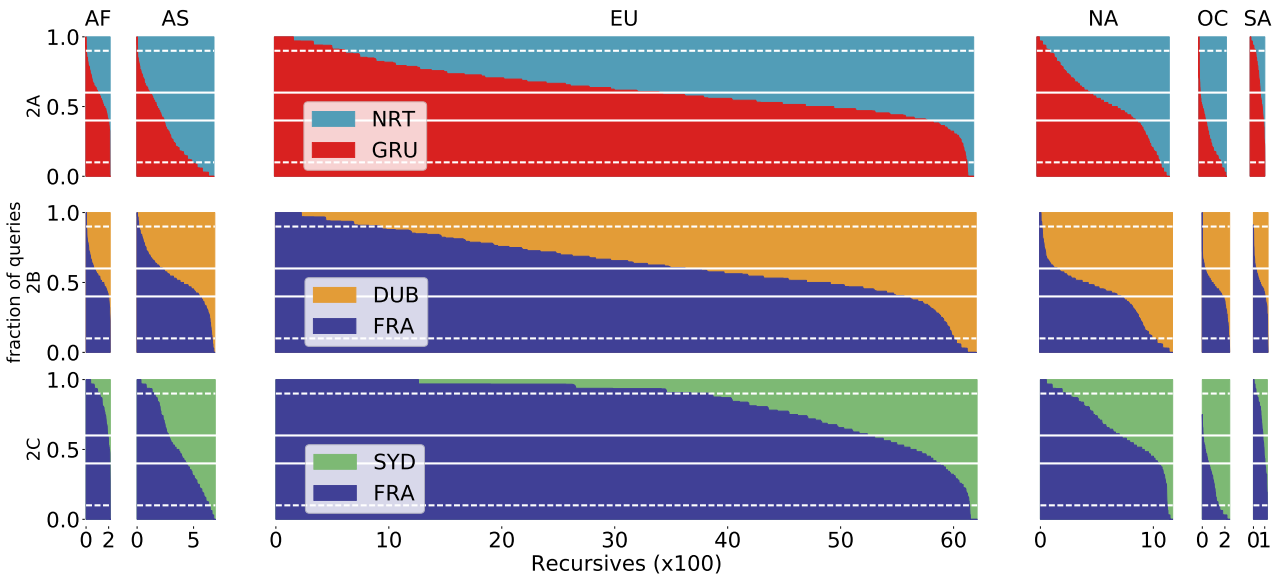

Figure 4: Recursive queries distribution for authoritative combinations 2A (top), 2B (center) and 2C (bottom). Solid and dotted horizontal lines mark VPs with weak and strong preference towards an authoritative.

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 (51ms) and always sees most queries.

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.* [29], 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 (VPs—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 50ms between the authoritatives. Based on our observations we define two thresholds for recursives preference: (i) a *weak* preference if the recursive sends at least 60% of its queries to one authoritative (solid lines in Figure 4); and (ii) a *strong* preference if at least 90% of queries go to one authoritative (dotted lines in Figure 4).

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. (We show in Figure 10 that recursives with a weak preference develop a stronger preference the longer they query the authoritatives. See Appendix C for more.)

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. (We have generated Figure 4 using data collected at Ripe Atlas probes (CL in Figure 1). We produce Figure 8 by analyzing pcap data collected at the authoritatives (AT in Figure 1) for the same measurements. See Appendix A for more.).

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

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, OC: Oceania, NA: North America, SA: South America)

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

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 13.9ms lower latency than DUB). In contrast, recursives in Asia distribute queries nearly equally, in spite of a similar difference in latency (AS VPs see 20.3ms difference). We conclude that *preferences based on RTT decrease when authoritatives are far away* (when they have large median RTT). 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 [2] caches latency for 10 minutes, and Unbound caches it for about 15 minutes [26]. 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;

but instead of a 2-minute interval between queries, we probe every 5, 10, 15, and 30 minutes. We choose

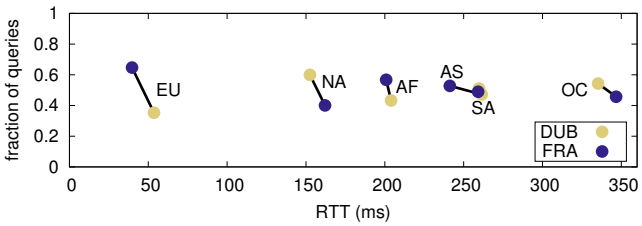

Figure 5: RTT sensitivity of 2B

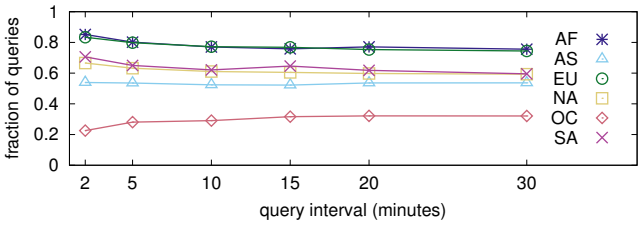

Figure 6: Fraction of queries to FRA (remainder go to SYD, configuration 2C), as query interval varies from 2 to 30 minutes.

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 we have analyzed recursive behavior in a measurement setup (§4) we now want to validate the results by looking at DNS traffic of real-life deployments of the root zone and the ccTLD `.nl`.

Root: We use DITL-2017 [7] 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 TLD: the picture slightly changes for queries to a country-code TLD. In the bottom plot of Figure 7 we plot the distribution of queries to 4 out of 8 .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.

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.* [29] 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 seen across the 9,000 vantage points, and by all users of the roots and .nl.

Kührer *et al.* [12] 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.

Also close to our work, Ager *et al.* [1] examine recursive resolution at 50 ISPs and Google Public DNS and OpenDNS. Our study considers many more recursives (more than 9000 locations in RIPE Atlas), and we focus on the role those recursives have in designing an authoritative server system.

Schomp *et al.* [23] consider the client-side of recursive resolvers. Unlike our work, they do not discuss implications for DNS operators.

Finally, other studies such as Castro *et al.* [6] have examined DNS traffic at the roots. They often use DITL data (as we do), but typical focus on client performance and balance of traffic across roots, 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. Because we have shown that most recursives will always send some queries to all authoritatives of a service, even if authoritatives employ large anycast networks for low latency, recursives will still send some queries to unicast sites sometimes, thus some queries will see higher latency. This cost is reduced for clients who preferentially choose a nearby site, but not eliminated. (Of course deployments of anycast to some authoratatives helps some queries, but not the tail.)

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. [24], seeing worse latency than they might otherwise.

Other Considerations: Other reasons motivate multiple authoritatives per service, or large use of anycast. Anycast is important to mitigate DDoS attacks [16]. In addition, standard practices recommend multiple authoritatives in different locations for fault tolerance [8].

However, for latency, prior work has shown that relatively few well-peered anycast sites can provide good global latency [22]. 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 for his 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 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, Informa-

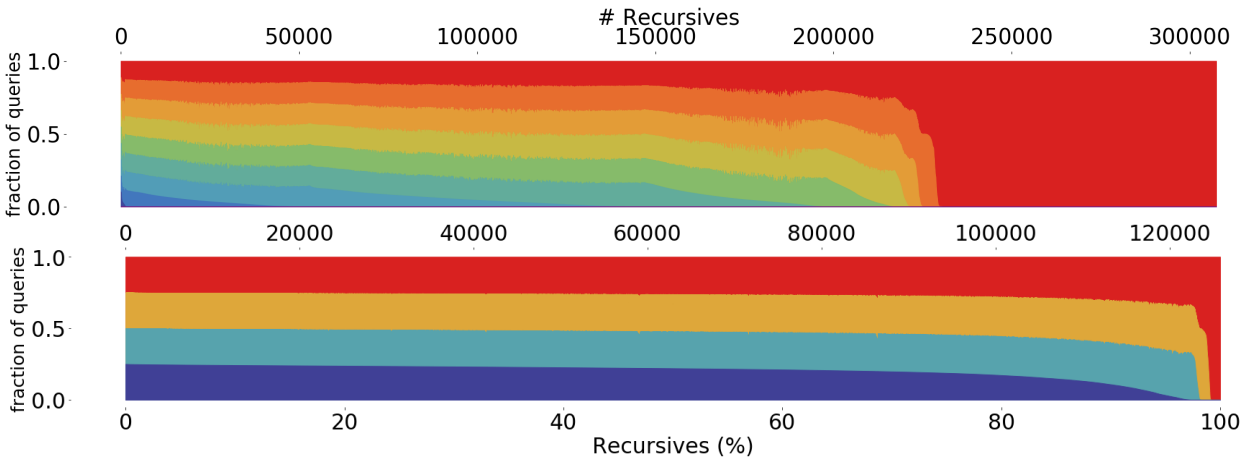

Figure 7: Distribution of queries of recursives with at least 250 queries across 10 out of 13 Root letters (top) and across 4 out of 8 name servers of .nl (bottom).

tion 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.

8. REFERENCES

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

[2] C. Almond. 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.

[3] V. Bajpai, , S. Eravuchira, J. Schönwälder, R. Kisteleki, and E. Aben. 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.

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

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

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

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

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

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

[10] Internet Assigned Numbers Authority (IANA). Root Files. https://www.iana.org/domains/root/files, 2017.

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

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

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

[14] P. Mockapetris. Domain names - concepts and facilities. RFC 1034 (INTERNET STANDARD), Nov. 1987.

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

[16] G. C. M. Moura, R. de O. Schmidt, J. Heidemann, W. B. de Vries, M. Müller, L. Wei, and C. Hesselman. Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. In *Proceedings of the 2016 ACM Conference on*

Internet Measurement Conference, pages 255–270, Oct. 2016.

[17] M. Müller, G. C. M. Moura, R. de O. Schmidt, and J. Heidemann. 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/dns/, May 2017.

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

[19] RIPE NCC. RIPE Atlas measurement ids. https://atlas.ripe.net/measurements/ID, 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.

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

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

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

[23] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman. On measuring the client-side dns infrastructure. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 77–90. ACM, Oct. 2013.

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

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

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

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

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

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

APPENDIX

A. CLIENT SELECTION FROM THE VIEW OF AUTHORITATIVE SERVERS

The question of how resolvers choose authoritatives in the wild can be measured from the *clients* (CL in Figure 1) and from the *authoritative servers* themselves (AT in that figure). Each method provides a slightly different perspective. In §4.3 we looked at data from the client's perspetvie to determine exactly which AT responded to each CL. This method does not make clear which R_n was used in the process. However, this client perspective represents the "user experience"—how users or applications will observe in practice the effects of choices of resolvers.

In this appendix we add data from the *authoriative servers*. This perspective allows us to observe which R_n chose which AT but does not allow to determine which CL issued each query, since they may be behind MI and even using multiple resolvers. This authoritative-side evaluation allows us to check our results from the client perspective hold with a different direction of analysis.

To do that, we evaluate the same measurements collected at each authoritative (available at [17]) and produce, per recursive resolver IP address (R_n), a distribution of queries across all authoritatives. The results is shown in Figure 8. We show only data for clients that send at least five queries during one measurement.

This authoritative-side analysis shows that our client-side results are not biased based on the location of their observation. Regardless of where it is measured (from the authoritative or client's point of view) most resolvers will choose authoritatives in the wild similarly, eventually selecting all possible authoritatives.

B. NUMBER OF AUTHORITATIVES PER TLD

The goal of our paper is to consider how DNS providers can configure their DNS services using replication with IP anycast (multiple sites per authoritiative server) and multiple NS records (multiple authoritative servers).

However, there is no consensus on of an optimal number of *ATs* for a TLD or DNS provider. For example, the Root DNS employs 13 authoritatives (or root server letters) with more than 500 anycast sites. The `.com` zone has a similar scope. The `.nl` zone, on the other hand, uses 8 authoritatives, some of which are anycast and some unicast.

We next briefly look across all top-level domains (TLDs) to see how many authoritatives they use. For this study we analyze the root zone file [10] on 2017-05-04, counting the number of authoritative records (NS) for each

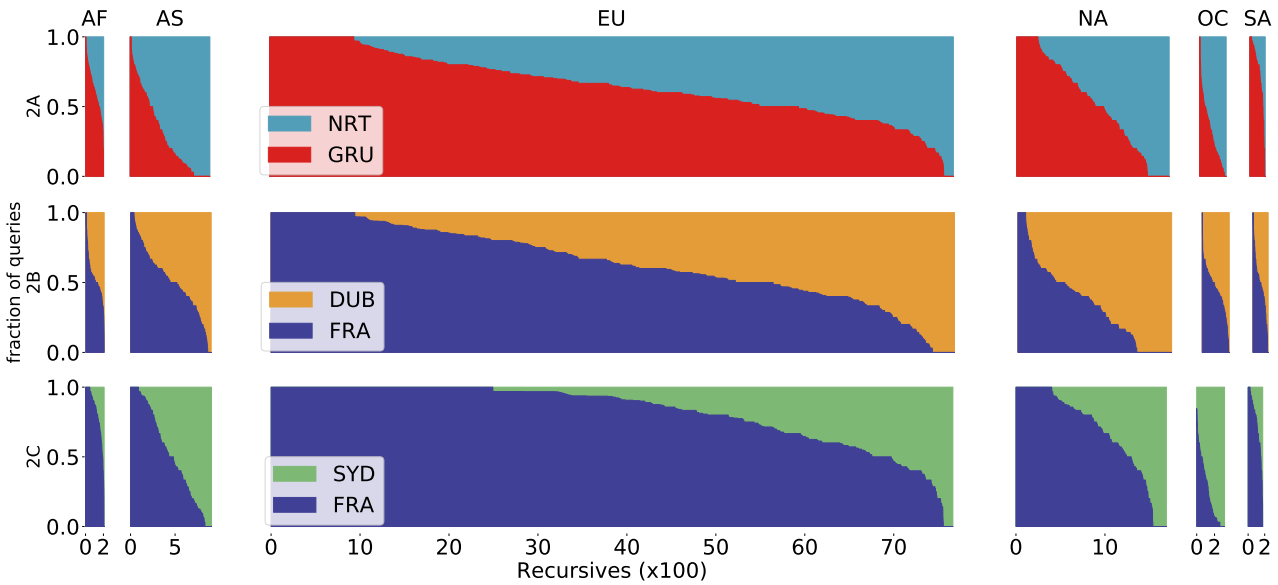

Figure 8: Recursive queries distribution for authoritative combinations 2A (top), 2B (center) and 2C (bottom). Measured at the authoritative and including only recursives that sent at least 5 queries.

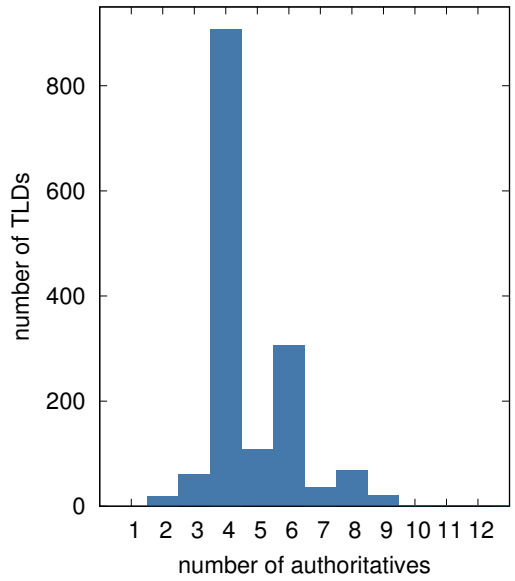

Figure 9: Number of authoritatives of TLDs in the root zone (2017-05-04).

TLD. Figure 9 shows the distribution. We can see that the majority of TLDs employ 4 NS records, in regardless of using anycast or unicast.

We also intent as future work to investigate the choice of multiple records in more details, as the deployment of IP anycast on these records.

C. PREFERENCE OF RECURSIVES OVER TIME

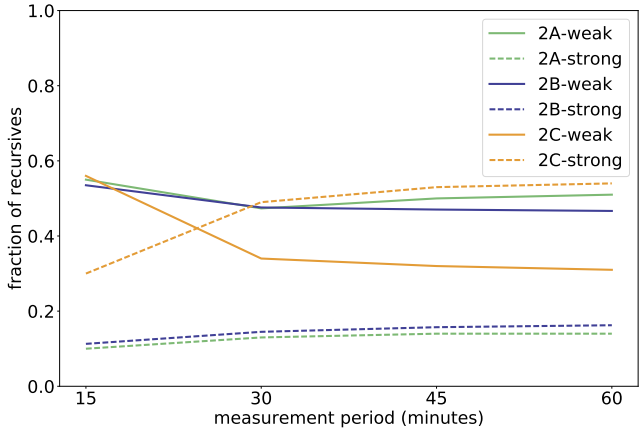

Figure 10: Fraction of recursives that have a weak (solid line) or a strong (dotted line) depending on the length of the measurement.

We show in §4.1 that recursives query all authoritatives fast, and in §4.2 that they develop a preference towards faster responding authoritatives. In this appendix we analyze if this preference changes over time.

Figure 9 shows the fraction of recursives for 2A – 2C that have a weak and strong preference with solid and dotted lines for measurements over a period of 15, 30, 45 and 60 minutes. In general, recursives with a weak preference develop a stronger preference over time, especially after 30 minutes. Therefore, operators can expect that the longer recursives with a preference send queries, the stronger they will prefer one of their authoritatives.