DNSOP Working Group
Internet-Draft
Intended status: Informational
Expires: October 10, 2020

G. Moura
SIDN Labs/TU Delft
W. Hardaker
J. Heidemann
USC/Information Sciences Institute
M. Davids
SIDN Labs

April 08, 2020

Considerations for Large Authoritative DNS Servers Operators
draft-moura-dnsop-authoritative-recommendations-07

Abstract

This document summarizes recent research work exploring Domain Name System (DNS) configurations and offers specific, tangible considerations to operators for configuring authoritative servers.

It is possible that the considerations presented in this document could be applicable in a wider context, such as for any stateless/ short-duration, anycasted service.

This document is not an IETF consensus document: it is published for informational purposes.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 10, 2020.

Table of Contents

1. Introduction 2
2. Background 3
3. C1: Use anycast in every authoritative for better load distribution 4
4. C2: Routing can matter more than locations 6
5. C3: Collecting anycast catchment maps to improve design . . . 7
6. C4: When under stress, employ two strategies 8
7. C5: Consider longer time-to-live values whenever possible . . 10
8. Security considerations 12
9. Privacy Considerations 12
10. IANA considerations 12
11. Acknowledgements 13
12. References 13
12.1. Normative References 13
12.2. Informative References 14
Authors' Addresses 16

1. Introduction

This document summarizes recent research work exploring DNS configurations and offers specific tangible considerations to DNS authoritative server operators (DNS operators hereafter). The considerations (C1-C5) presented in this document are backed by previous research work, which used wide-scale Internet measurements upon which to draw their conclusions. This document describes the key engineering options, and points readers to the pertinent papers for details and other research works related to each consideration here presented.

These considerations are designed for operators of "large" authoritative servers. In this context, "large" authoritative servers refers to those with a significant global user population, like top-level domain (TLD) operators, run by a single or multiple operators. These considerations may not be appropriate for smaller domains, such as those used by an organization with users in one city or region, where goals such as uniform low latency are less strict.

It is likely that these considerations might be useful in a wider context, such as for any stateless/short-duration, anycasted service. Because the conclusions of the studies don't verify this fact, the wording in this document discusses DNS authoritative services only. This document is not an IETF consensus document: it is published for informational purposes.

2. Background

The DNS as main two types of DNS servers: authoritative servers and recursive resolvers. Figure 1 shows their relationship. An authoritative server (ATn in Figure 1) knows the content of a DNS zone from local knowledge, and thus can answer queries about that zone without needing to query other servers [RFC2181]. A recursive resolver (Re1-Re3) is a program that extracts information from name servers in response to client requests [RFC1034]. A client (stub in Figure 1) refers to stub resolver [RFC1034] that is typically located within the client software.

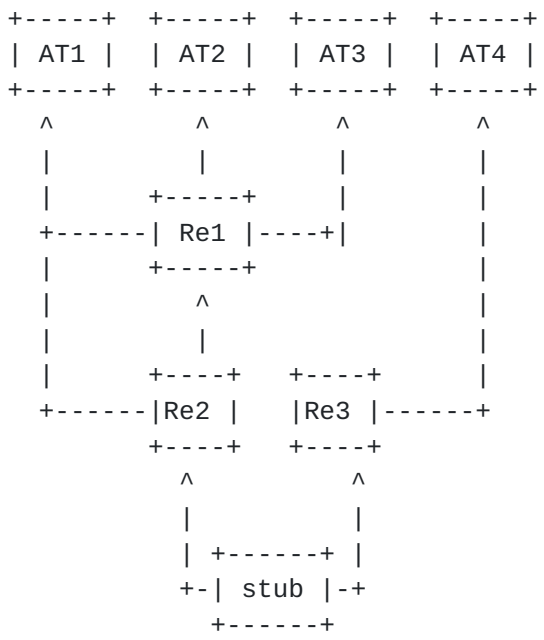

Figure 1: Relationship between recursive resolvers (Re) and authoritative name servers (ATn)

DNS queries/responses contribute to a user's perceived perceived latency and affect user experience [Sigla2014], and the DNS system has been subject to repeated Denial of Service (DoS) attacks (for example, in November 2015 [Moura16b]) in order to degrade user experience.

To reduce latency and improve resiliency against DoS attacks, DNS uses several types of server replication. Replication at the authoritative server level can be achieved with (i) the deployment of multiple servers for the same zone [RFC1035] (AT1--AT4 in Figure 1), (ii) the use of IP anycast [RFC1546][RFC4786][RFC7094] that allows the same IP address to be announced from multiple locations (each of them referred to as an anycast instance [RFC8499]) and (iii) by using load balancers to support multiple servers inside a single (potentially anycasted) instance. As a consequence, there are many possible ways an authoritative DNS provider can engineer its production authoritative server network, with multiple viable choices and no single optimal design.

In the next sections we cover specific considerations (C1-C5) for large authoritative DNS server operators.

3. C1: Use anycast in every authoritative for better load distribution

Authoritative DNS servers operators announce their authoritative servers as NS records[RFC1034]. Different authoritatives for a given zone should return the same content, typically by staying synchronized using DNS zone transfers (AXFR[RFC5936] and IXFR[RFC1995]) to coordinate the authoritative zone data to return to their clients.

DNS heavily relies upon replication to support high reliability, capacity and to reduce latency [Moura16b]. DNS has two complementary mechanisms to replicate the service. First, the protocol itself supports nameserver replication of DNS service for a DNS zone through the use of multiple nameservers that each operate on different IP addresses, listed by a zone's NS records. Second, each of these network addresses can run from multiple physical locations through the use of IP anycast[RFC1546][RFC4786][RFC7094], by announcing the same IP address from each instance and allowing Internet routing (BGP[RFC4271]) to associate clients with their topologically nearest anycast instance. Outside the DNS protocol, replication can be achieved by deploying load balancers at each physical location. Nameserver replication is recommended for all zones (multiple NS records), and IP anycast is used by most large zones such as the DNS Root, most top-level domains[Moura16b] and large commercial enterprises, governments and other organizations.

Most DNS operators strive to reduce latency for users of their service. However, because they control only their authoritative servers, and not the recursive resolvers communicating with those servers, it is difficult to ensure that recursives will be served by the closest authoritative server. Server selection is up to the recursive resolver's software implementation, and different software

vendors and releases employ different criteria to chose which authoritative servers with which to communicate.

Knowing how recursives choose authoritative servers is a key step to better engineer the deployment of authoritative servers. [Mueller17b] evaluates this with a measurement study in which they deployed seven unicast authoritative name servers in different global locations and queried these authoritative servers from more than 9k RIPE authoritative server operators and their respective recursive resolvers.

In the wild, [Mueller17b] found that recursives query all available authoritative servers, regardless of the observed latency. But the distribution of queries tends to be skewed towards authoritatives with lower latency: the lower the latency between a recursive resolver and an authoritative server, the more often the recursive will send queries to that authoritative. These results were obtained by aggregating results from all vantage points and not specific to any vendor/version.

The hypothesis is that this behavior is a consequence of two main criteria employed by resolvers when choosing authoritatives: performance (lower latency) and diversity of authoritatives, where a resolver checks all authoritative servers to determine which is closer and to provide alternatives if one is unavailable.

For a DNS operator, this policy means that latency of all authoritatives (NS records) matter, so all must be similarly capable, since all available authoritatives will be queried by most recursive resolvers. Since unicast cannot deliver good latency worldwide (a unicast authoritative server in Europe will always have high latency to resolvers in California, for example, given its geographical distance), [Mueller17b] recommends to DNS operators that they deploy equally strong IP anycast in every authoritative server (i.e., on each NS record , in terms of number of instances and peering, and, consequently, to phase out unicast, so they can deliver good latency values to global clients. However, [Mueller17b] also notes that DNS operators should also take architectural considerations into account when planning for deploying anycast [RFC1546].

This consideration was deployed at the ".nl" TLD zone, which originally had seven authoritative severs (mixed unicast/anycast setup). In early 2018, .nl moved to a setup with 4 anycast authoritative name servers. This is not to say that .nl was the first - other zones, have been running anycast only authoritatives (e.g., .be since 2013). [Mueller17b] contribution is to show that unicast cannot deliver good latency worldwide, and that anycast has to be deployed to deliver good latency worldwide.

4. C2: Routing can matter more than locations

A common metric when choosing an anycast DNS provider or setting up an anycast service is the number of anycast instances[RFC4786], i.e., the number of global locations from which the same address is announced with BGP. Intuitively, one could think that more instances will lead to shorter response times.

However, this is not necessarily true. In fact, [Schmidt17a] found that routing can matter more than the total number of locations. They analyzed the relationship between the number of anycast instances and the performance of a service (latency-wise, round-trip time (RTT)) and measured the overall performance of four DNS Root servers. The Root DNS is implemented by 13 separate DNS services, each running on a different IP address, but sharing a common master data source: the root DNS zone. These are called the 13 DNS Root Letter Services just the "Root Letters" for short), since each is assigned a letter from A to M and identified as $letter.root-servers.net.

In specific, [Schmidt17a] measured the performance of C, F, K and L root letters, from more than 7.9k RIPE Atlas probes (RIPE Atlas is a measurement platform with more than 12000 global devices - Atlas Probes - that provide vantage points that conduct Internet measurements, and its regularly used by researchers and operators [RipeAtlas15a] {{RipeAtlas19a}).

[Schmidt17a] found that C-Root, a smaller anycast deployment consisting of only 8 instances (they refer to anycast instance as anycast site), provided a very similar overall performance than that of the much larger deployments of K and L, with 33 and 144 instances respectively. The median RTT for C, K and L Root was between 30-32ms.

Given that Atlas has better coverage in Europe than other regions, the authors specifically analyzed results per region and per country (Figure 5 in [Schmidt17a]), and show that Atlas bias to Europe does not change the conclusion that location of anycast instances dominates latency.

[Schmidt17a] consideration for DNS operators when engineering anycast services is consider factors other than just the number of instances (such as local routing connectivity) when designing for performance. They showed that 12 instances can provide reasonable latency, given they are globally distributed and have good local interconnectivity. However, more instances can be useful for other reasons, such as when handling Denial-of-service (DoS) attacks [Moura16b].

5. C3: Collecting anycast catchment maps to improve design

An anycast DNS service may have several dozens or even more than one hundred locations (such as L-Root does). Anycast leverages Internet routing to distribute the incoming queries to a service's distributed anycast locations; in theory, BGP (the Internet's de facto routing protocol) forwards incoming queries to a nearby anycast location (in terms of BGP distance). However, usually queries are not evenly distributed across all anycast locations, as found in the case of L-Root [IcannHedge18].

Adding locations to an anycast service may change the load distribution across all locations. Given that BGP maps clients to locations, whenever a new location is announced, this new location may receive more or less traffic than it was engineered for, leading to suboptimal usage of the service or even stressing the new location while leaving others underutilized. This is a scenario that operators constantly face when expanding an anycast service. Besides, when setting up a new anycast service location, operators cannot directly estimate the query distribution among the locations in advance of enabling the new location.

To estimate the query loads across locations of an expanding service or a when setting up an entirely new service, operators need detailed anycast maps and catchment estimates (i.e., operators need to know which prefixes will be matched to which anycast instance). To do that, [Vries17b] developed a new technique enabling operators to carry out active measurements, using an open-source tool called Verfploeter (available at [VerfSrc]). Verfploeter maps a large portion of the IPv4 address space, allowing DNS operators to predict both query distribution and clients catchment before deploying new anycast instances. At the moment of this writing, Verfploeter still does not support IPv6.

[Vries17b] shows how this technique was used to predict both the catchment and query load distribution for the new anycast service of B-Root. Using two anycast instances in Miami (MIA) and Los Angeles (LAX) from the operational B-Root server, they sent ICMP echo packets to IP addresses to each IPv4 /24 on the Internet using a source address within the anycast prefix. Then, they recorded which instance the ICMP echo replies arrived at based on the Internet's BGP routing. This analysis resulted in an Internet wide catchment map. Weighting was then applied to the incoming traffic prefixes based on of 1 day of B-Root traffic (2017-04-12, DITL datasets [Ditl17]). The combination of the created catchment mapping and the load per prefix created an estimate predicting that 81.6% of the traffic would go to the LAX location. The actual value was 81.4% of traffic going to LAX, showing that the estimation was pretty close and the Verfploeter

technique was a excellent method of predicting traffic loads in advance of a new anycast instance deployment ([Vries17b] also uses the term anycast site to refer to anycast location).

Besides that, Verfploeter can also be used to estimate how traffic shifts among locations when BGP manipulations are executed, such as AS Path prepending that is frequently used by production networks during DDoS attacks. A new catchment mapping for each prepending configuration configuration: no prepending, and prepending with 1, 2 or 3 hops at each instance. Then, [Vries17b] shows that this mapping can accurately estimate the load distribution for each configuration.

An important operational takeaway from [Vries17b] is that DNS operators can make informed choices when engineering new anycast locations or when expending new ones by carrying out active measurements using Verfploeter in advance of operationally enabling the fully anycast service. Operators can spot sub-optimal routing situations early, with a fine granularity, and with significantly better coverage than using traditional measurement platforms such as RIPE Atlas.

To date, Verfploeter has been deployed on B-Root[Vries17b], on a operational testbed (Anycast testbed) [AnyTest], and on a large unnamed operator.

The consideration is therefore to deploy a small test Verfploeter-enabled platform in advance at a potential anycast locations may reveal the realizable benefits of using that location as an anycast interest, potentially saving significant financial and labor costs of deploying hardware to a new location that was less effective than as had been hoped.

6. C4: When under stress, employ two strategies

DDoS attacks are becoming bigger, cheaper, and more frequent [Moura16b]. The most powerful recorded DDoS attack to DNS servers to date reached 1.2 Tbps, by using IoT devices [Perlroth16]. Such attacks call for an answer for the following question: how should a DNS operator engineer its anycast authoritative DNS server react to the stress of a DDoS attack? This question is investigated in study [Moura16b] in which empirical observations are grounded with the following theoretical evaluation of options.

An authoritative DNS server deployed using anycast will have many server instances distributed over many networks. Ultimately, the relationship between the DNS provider's network and a client's ISP will determine which anycast instance will answer queries for a given client, given that BGP is the protocol that maps clients to specific

anycast instances by using routing information [RF:KDar02]. As a consequence, when an anycast authoritative server is under attack, the load that each anycast instance receives is likely to be unevenly distributed (a function of the source of the attacks), thus some instances may be more overloaded than others which is what was observed analyzing the Root DNS events of Nov. 2015 [Moura16b]. Given the fact that different instances may have different capacity (bandwidth, CPU, etc.), making a decision about how to react to stress becomes even more difficult.

In practice, an anycast instance under stress, overloaded with incoming traffic, has two options:

- It can withdraw or pre-prepend its route to some or to all of its neighbors, perform other traffic shifting tricks (such as reducing the propagation of its announcements using BGP communities[RFC1997]) which shrinks portions of its catchment), use FlowSpec [RFC5575] or other upstream communication mechanisms to deploy upstream filtering. The goals of these techniques is to perform some combination of shifting of both legitimate and attack traffic to other anycast instances (with hopefully greater capacity) or to block the traffic entirely.

- Alternatively, it can be become a degraded absorber, continuing to operate, but with overloaded ingress routers, dropping some incoming legitimate requests due to queue overflow. However, continued operation will also absorb traffic from attackers in its catchment, protecting the other anycast instances.

[Moura16b] saw both of these behaviors in practice in the Root DNS events, observed through instance reachability and route-trip time (RTTs). These options represent different uses of an anycast deployment. The withdrawal strategy causes anycast to respond as a waterbed, with stress displacing queries from one instance to others. The absorption strategy behaves as a conventional mattress, compressing under load, with some queries getting delayed or dropped.

Although described as strategies and policies, these outcomes are the result of several factors: the combination of operator and host ISP routing policies, routing implementations withdrawing under load, the nature of the attack, and the locations of the instances and the attackers. Some policies are explicit, such as the choice of local-only anycast instances, or operators removing an instance for maintenance or modifying routing to manage load. However, under stress, the choices of withdrawal and absorption can also be results that emerge from a mix of explicit choices and implementation details, such as BGP timeout values.

[Moura16b] speculates that more careful, explicit, and automated management of policies may provide stronger defenses to overload. For DNS operators, that means that besides traditional filtering, two other options are available (withdraw/prepend/communities or isolate instances), and the best choice depends on the specifics of the attack.

Note that this consideration refers to the operation of one anycast service, i.e., one anycast NS record. However, DNS zones with multiple authoritative anycast servers may expect load to spill from one anycast server to another, as resolvers switch from authoritative to authoritative when attempting to resolve a name [Mueller17b].

7. C5: Consider longer time-to-live values whenever possible

Caching is the cornerstone of good DNS performance and reliability. A 15 ms response to a new DNS query is fast, but a 1 ms cache hit to a repeat query is far faster. Caching also protects users from short outages and can mute even significant DDoS attacks [Moura18b].

DNS record TTLs (time-to-live values) directly control cache duration [RFC1034][RFC1035] and, therefore, affect latency, resilience, and the role of DNS in CDN server selection. Some early work modeled caches as a function of their TTLs [Jung03a], and recent work examined their interaction with DNS[Moura18b], but no research provides considerations about what TTL values are good. With this goal Moura et. al. [Moura19a] carried out a measurement study investigating TTL choices and its impact on user experience in the wild, and not focused on specific resolvers (and their caching architectures), vendors, or setups.

First, they identified several reasons why operators/zone owners may want to choose longer or shorter TTLs:

- Longer TTL leads to longer caching, which results in faster responses, given that cache hits are faster than cache misses in resolvers. [Moura19a] shows that the increase in the TTL for .uy TLD from 5 minutes (300s) to 1 day (86400s) reduced the latency from 15k Atlas vantage points significantly: the median RTT went from 28.7ms to 8ms, while the 75%ile decreased from 183ms to 21ms.

- Longer caching results in lower DNS traffic: authoritative servers will experience less traffic if TTLs are extended, given that repeated queries will be answered by resolver caches.

- Longer caching results in lower cost if DNS is metered: some DNS-As-A-Service providers charges are metered, with a per query cost (often added to a fixed monthly cost).

- Longer caching is more robust to DDoS attacks on DNS: DDoS attacks on a DNS service provider harmed several prominent websites [Perlroth16]. Recent work has shown that DNS caching can greatly reduce the effects of DDoS on DNS, provided caches last longer than the attack [Moura18b].

- Shorter caching supports operational changes: An easy way to transition from an old server to a new one is to change the DNS records. Since there is no method to remove cached DNS records, the TTL duration represents a necessary transition delay to fully shift to a new server, so low TTLs allow more rapid transition. However, when deployments are planned in advance (that is, longer than the TTL), then TTLs can be lowered ''just-before'' a major operational change, and raised again once accomplished.

- Shorter caching can help with a DNS-based response to DDoS attacks: Some DDoS-scrubbing services use DNS to redirect traffic during an attack. Since DDoS attacks arrive unannounced, DNS-based traffic redirection requires the TTL be keptquite low at all times to be ready to respond to a potential attack.

- Shorter caching helps DNS-based load balancing: Many large services use DNS-based load balancing. Each arriving DNS request provides an opportunity to adjust load, so short TTLs may be desired to react more quickly to traffic dynamics. (Although many recursive resolvers have minimum caching times of tens of seconds, placing a limit on agility.)

As such, choice of TTL depends in part on external factors so no single recommendation is appropriate for all. Organizations must weigh these trade-offs to find a good balance. Still, some guidelines can be used when choosing TTLs:

- For general users, [Moura19a] recommends longer TTLs, of at least one hour, and ideally 8, 12, or 24 hours. Assuming planned maintenance can be scheduled at least a day in advance, long TTLs have little cost.

- For TLD operators: TLD operators that allow public registration of domains (such as most ccTLDs and .com, .net, .org) host, in their zone files, NS records (and glues if in-bailiwick) of their respective domains. [Moura19a] shows that most resolvers will use TTL values provided by the child delegations, but some will choose the TTL provided by the parents. As such, similarly to general users, [Moura19a] recommends longer TTLs for NS records of their delegations (at least one hour, preferably more).

- Users of DNS-based load balancing or DDoS-prevention may require short TTLs: TTLs may be as short as 5 minutes, although 15 minutes may provide sufficient agility for many operators. Shorter TTLs here help agility; they are are an exception to the consideration for longer TTLs.

- Use A/AAAA and NS records: TTLs of A/AAAA records should be shorter or equal to the TTL for NS records for in-bailiwick authoritative DNS servers, given that the authors [Moura19a] found that, for such scenarios, once NS record expires, their associated A/AAAA will also be updated (glue is sent by the parents). For out-of-bailiwick servers, A and NS records are usually cached independently, so different TTLs, if desired, will be effective. In either case, short A and AAAA records may be desired if DDoS-mitigation services are an option.

8. Security considerations

As this document discusses applying research results to operational deployments, there are no further security considerations, other than the ones mentioned in the normative references. Most of the considerations affect mostly operational practice, though a few do have security related impacts, which we'll summarize at high level.

Specifically, C4 discusses a few strategies to employ when a service is under stress, providing operators with additional guidance when handling denial of service attacks.

Similarly, C5 identifies both the operational and security benefits to using longer time-to-live values.

9. Privacy Considerations

This document does not add any practical new privacy issues, aside from possible benefits in deploying longer TTLs as suggested in C5. Longer TTLs may help preserve a user's privacy by reducing the number of requests that get transmitted in both thec lient-to-resolver and resolver-to-authoritative cases.

DNS privacy is currently under active study, and future research efforts by multiple organizations may produce more guidance in this area.

10. IANA considerations

This document has no IANA actions.

11. Acknowledgements

This document is a summary of the main considerations of six research works referred in this document. As such, they were only possible thanks to the hard work of the authors of these research works.

- Ricardo de O. Schmidt
- Wouter B de Vries
- Moritz Mueller
- Lan Wei
- Cristian Hesselman
- Jan Harm Kuipers
- Pieter-Tjerk de Boer
- Aiko Pras

We would like also to thank the various reviewers of different versions of this draft: Duane Wessels, Joe Abley, Toema Gavrichenkov, John Levine, Michael StJohns, Kristof Tuyteleers, Stefan Ubbink, Klaus Darilion and Samir Jafferali, and comments provided at the IETF DNSOP session (IETF104).

Besides those, we would like thank those who have been individually thanked in each research work, RIPE NCC and DNS OARC for their tools and datasets used in this research, as well as the funding agencies sponsoring the individual research works.

12. References

12.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, DOI 10.17487/RFC1546, November 1993, <https://www.rfc-editor.org/info/rfc1546>.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <https://www.rfc-editor.org/info/rfc1997>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786, December 2006, <https://www.rfc-editor.org/info/rfc4786>.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, DOI 10.17487/RFC5575, August 2009, <https://www.rfc-editor.org/info/rfc5575>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/info/rfc5936>.

[RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <https://www.rfc-editor.org/info/rfc7094>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

12.2. Informative References

[AnyTest] Schmidt, R., "Anycast Testbed", December 2018, <http://www.anycast-testbed.com/>.

[Ditl17] OARC, D., "2017 DITL data", October 2018, <https://www.dns-oarc.net/oarc/data/ditl/2017>.

[IcannHedge18]
ICANN, ., "DNS-STATS - Hedgehog 2.4.1", October 2018, <http://stats.dns.icann.org/hedgehog/>.

[Jung03a] Jung, J., Berger, A., and H. Balakrishnan, "Modeling TTL-based Internet caches", ACM 2003 IEEE INFOCOM, DOI 10.1109/INFCOM.2003.1208693, July 2003, <http://www.ieee-infocom.org/2003/papers/11_01.PDF>.

[Moura16b]
Moura, G., Schmidt, R., Heidemann, J., Mueller, M., Wei, L., and C. Hesselman, "Anycast vs DDoS Evaluating the November 2015 Root DNS Events.", ACM 2016 Internet Measurement Conference, DOI /10.1145/2987443.2987446, October 2016, <https://www.isi.edu/~johnh/PAPERS/Moura16b.pdf>.

[Moura18b]
Moura, G., Heidemann, J., Mueller, M., Schmidt, R., and M. Davids, "When the Dike Breaks: Dissecting DNS Defenses During DDos", ACM 2018 Internet Measurement Conference, DOI 10.1145/3278532.3278534, October 2018, <https://www.isi.edu/~johnh/PAPERS/Moura18b.pdf>.

[Moura19a]
Moura, G., Heidemann, J., Schmidt, R., and W. Hardaker, "Cache Me If You Can: Effects of DNS Time-to-Live", ACM 2019 Internet Measurement Conference, DOI 10.1145/3355369.3355568, October 2019, <https://www.isi.edu/~johnh/PAPERS/Moura19b.pdf>.

[Mueller17b]
Mueller, M., Moura, G., Schmidt, R., and J. Heidemann, "Recursives in the Wild- Engineering Authoritative DNS Servers.", ACM 2017 Internet Measurement Conference, DOI 10.1145/3131365.3131366, October 2017, <https://www.isi.edu/%7ejohnh/PAPERS/Mueller17b.pdf>.

[Perlroth16]
Perlroth, N., "Hackers Used New Weapons to Disrupt Major Websites Across U.S.", October 2016, <https://www.nytimes.com/2016/10/22/business/internet-problems-attack.html>.

[RipeAtlas15a]
Staff, R., "RIPE Atlas A Global Internet Measurement Network", September 2015, <http://ipj.dreamhosters.com/wp-content/uploads/issues/2015/ipj18-3.pdf>.

[RipeAtlas19a]
NCC, R., "Ripe Atlas - RIPE Network Coordination Centre", September 2019, <https://atlas.ripe.net/>.

[Schmidt17a]
Schmidt, R., Heidemann, J., and J. Kuipers, "Anycast Latency - How Many Sites Are Enough. In Proceedings of the Passive and Active Measurement Workshop", PAM Passive and Active Measurement Conference, March 2017, <https://www.isi.edu/%7ejohnh/PAPERS/Schmidt17a.pdf>.

[Sigla2014]
Singla, A., Chandrasekaran, B., Godfrey, P., and B. Maggs, "The Internet at the speed of light. In Proceedings of the 13th ACM Workshop on Hot Topics in Networks (Oct 2014)", ACM Workshop on Hot Topics in Networks, October 2014, <http://speedierweb.web.engr.illinois.edu/cspeed/papers/hotnets14.pdf>.

[VerfSrc] Vries, W., "Verfploeter source code", November 2018, <https://github.com/Woutifier/verfploeter>.

[Vries17b]
Vries, W., Schmidt, R., Hardaker, W., Heidemann, J., Boer, P., and A. Pras, "Verfploeter - Broad and Load-Aware Anycast Mapping", ACM 2017 Internet Measurement Conference, DOI 10.1145/3131365.3131371, October 2017, <https://www.isi.edu/%7ejohnh/PAPERS/Vries17b.pdf>.

Authors' Addresses

Giovane C. M. Moura
SIDN Labs/TU Delft
Meander 501
Arnhem 6825 MD
The Netherlands

Phone: +31 26 352 5500
Email: giovane.moura@sidn.nl

Wes Hardaker
USC/Information Sciences Institute
PO Box 382
Davis 95617-0382
U.S.A.

Phone: +1 (530) 404-0099
Email: ietf@hardakers.net

John Heidemann
USC/Information Sciences Institute
4676 Admiralty Way
Marina Del Rey 90292-6695
U.S.A.

Phone: +1 (310) 448-8708
Email: johnh@isi.edu

Marco Davids
SIDN Labs
Meander 501
Arnhem 6825 MD
The Netherlands

Phone: +31 26 352 5500
Email: marco.davids@sidn.nl