DNSOP Working Group
Internet-Draft
Intended status: Informational
Expires: August 23, 2021

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

February 19, 2021

Considerations for Large Authoritative DNS Servers Operators
draft-moura-dnsop-authoritative-recommendations-08

Abstract

Recent research work has explored the deployment characteristics and configuration of the Domain Name System (DNS). This document summarizes the conclusions from these research efforts and offers specific, tangible advice to operators when configuring authoritative DNS servers.

It is possible that the results presented in this document could be applicable in a wider context than just the DNS protocol, as some of the results may generically apply to any stateless/short-duration, anycasted service.

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

Status of This Memo

Table of Contents

1. Introduction

This document summarizes recent research work that explored the deployed DNS configurations and offers derived, specific tangible advice to DNS authoritative server operators (DNS operators hereafter). The considerations (C1--C5) presented in this document are backed by published research work, which used wide-scale Internet measurements to draw their conclusions. This document summarizes the research results and describes the resulting key engineering options. In each section, it points readers to the pertinent publications where additional details are presented.

These considerations are designed for operators of "large" authoritative DNS servers. In this context, "large" authoritative

servers refers to those with a significant global user population, like top-level domain (TLD) operators, run by either a single or multiple operators. Typically these networks are deployed on wide anycast networks [RFC1546]. These considerations may not be appropriate for smaller domains, such as those used by an organization with users in one unicast network, or in one city or region, where operational goals such as uniform, global low latency are less required.

It is possible that the results presented in this document could be applicable in a wider context than just the DNS protocol, as some of the results may generically apply to any stateless/short-duration, anycasted service. Because the conclusions of the reviewed studies don't measure smaller networks, the wording in this document concentrates solely on disusing large-scale DNS authoritative services only.

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

2. Background

The DNS has main two types of DNS servers: authoritative servers and recursive resolvers, shown by a representational deployment model in Figure 1. An authoritative server (shown as AT1--AT4 in Figure 1) knows the content of a DNS zone, and is responsible for answering queries about that zone. It runs using local (possibly automatically updated) copies of the zone and does not need to query other servers [RFC2181] in order to answer requests. A recursive resolver (Re1--Re3) is a server that iteratively queries authoritative and other servers to answer queries received from client requests [RFC1034]. A client typically employs a software library called a stub resolver (stub in Figure 1) to issue its query to the upstream recursive resolvers [RFC1034].

```
+-----+  +-----+  +-----+  +-----+
| AT1 |  | AT2 |  | AT3 |  | AT4 |
+-----+  +-----+  +-----+  +-----+
  ^         ^        ^        ^
  |         |        |        |
  |      +-----+     |        |
  +------| Re1 |----+|        |
  |      +-----+              |
  |         ^                 |
  |         |                 |
  |      +----+   +----+      |
  +------|Re2 |   |Re3 |------+
         +----+   +----+
           ^          ^
           |          |
           | +------+ |
           +-| stub |-+
             +------+
```

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

DNS queries issued by a client contribute to a user's perceived perceived latency and affect user experience [Sigla2014] depending on how long it takes for responses to be returned. The DNS system has been subject to repeated Denial of Service (DoS) attacks (for example, in November 2015 [Moura16b]) in order to specifically degrade user experience.

To reduce latency and improve resiliency against DoS attacks, the DNS uses several types of service 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 referred to as an "anycast instance" [RFC8499]) and (iii) the use of 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 necessarily single optimal design.

In the next sections we cover the specific consideration (C1--C5) for conclusions drawn within the academic papers about large authoritative DNS server operators.

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

Authoritative DNS server operators announce their service using NS records[RFC1034]. Different authoritative servers for a given zone should return the same content; typically they stay synchronized using DNS zone transfers (AXFR[RFC5936] and IXFR[RFC1995]), coordinating the zone data they all return to their clients.

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

Most DNS operators strive to reduce service latency for users. However, because they only have control over their authoritative servers, and not over the client recursive resolvers, 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 vendors and even different releases employ different criteria to chose the authoritative servers with which to communicate.

Understanding how recursive resolvers choose authoritative servers is a key step in improving the effectiveness of authoritative server deployments. To measure and evaluate server deployments, [Mueller17b] deployed seven unicast authoritative name servers in different global locations and then queried them from more than 9000 RIPE authoritative server operators and their respective recursive resolvers.

[Mueller17b] found that recursive resolvers in the wild 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 server. These results were obtained by aggregating results from all of the vantage points and were not specific to any specific vendor or version.

The authors believe this behavior is a consequence of combining the two main criteria employed by resolvers when selecting authoritative servers: resolvers regularly check all listed authoritative servers in an NS set to determine which is closer (the least latent) and when one isn't available selects one of the alternatives.

For an authoritative DNS operator, this result means that the latency of all authoritative servers (NS records) matter, so they all must be similarly capable -- all available authoritatives will be queried by most recursive resolvers. Unicasted services, unfortunately, cannot deliver good latency worldwide (a unicast authoritative server in Europe will always have high latency to resolvers in California and Australia, for example, given its geographical distance). [Mueller17b] recommends that DNS operators deploy equally strong IP anycast instances for every authoritative server (i.e., for each NS record). Each large authoritative DNS server provider should phase out their usage of unicast and deploy a well engineered number of anycast instances with good peering strategies so they can provide good latency to their global clients.

As a case study, the ".nl" TLD zone was originally served on seven authoritative servers with a mixed unicast/anycast setup. In early 2018, .nl moved to a setup with 4 anycast authoritative servers.

[Mueller17b]'s contribution to DNS service engineering shows that because unicast cannot deliver good latency worldwide, anycast needs to be used to provide a low latency service worldwide.

4. C2: Routing can matter more than locations

When selecting an anycast DNS provider or setting up an anycast service, choosing the best number of anycast instances[RFC4786] to deploy is a challenging problem. Selecting where and how many global locations to announce from using BGP is tricky. Intuitively, one could naively think that the more instances the better and simply "more" will always lead to shorter response times.

This is not necessarily true, however. In fact, [Schmidt17a] found that proper route engineering can matter more than the total number of locations. They analyzed the relationship between the number of anycast instances and service performance (measuring latency of the round-trip time (RTT)), measuring the overall performance of four DNS Root servers. The Root DNS servers are implemented by 12 separate

organizations serving the DNS root zone at 13 different IPv4/IPv6 address pairs.

The results documented in [Schmidt17a] measured the performance of the {c,f,k,l}.root-servers.net (hereafter, "C", "F", "K" and "L") servers from more than 7.9k RIPE Atlas probes. RIPE Atlas is a Internet measurement platform with more than 12000 global vantage points called "Atlas Probes" -- it is used regularly by both researchers and operators [RipeAtlas15a] [RipeAtlas19a].

[Schmidt17a] found that the C server, a smaller anycast deployment consisting of only 8 instances, provided very similar overall performance in comparison to the much larger deployments of K and L, with 33 and 144 instances respectively. The median RTT for C, K and L root server were all between 30-32ms.

Because RIPE Atlas is known to have better coverage in Europe than other regions, the authors specifically analyzed the results per region and per country (Figure 5 in [Schmidt17a]), and show that known Atlas bias toward Europe does not change the conclusion that properly selected anycast locations is more important to latency than the number of sites.

The important conclusion of [Schmidt17a] is that when engineering anycast services for performance, factors other than just the number of instances (such as local routing connectivity) must be considered. They showed that 12 instances can provide reasonable latency, assuming they are globally distributed and have good local interconnectivity. However, additional instances can still 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 be deployed from anywhere from several locations to hundreds of locations (for example, l.root-servers.net has over 150 anycast instances at the time this was written). Anycast leverages Internet routing to distribute incoming queries to a service's hop-nearest distributed anycast locations. However, usually queries are not evenly distributed across all anycast locations, as found in the case of L-Root [IcannHedge18].

Adding locations to or removing locations from a deployed anycast network changes the load distribution across all of its locations. When a new location is announced by BGP, locations may receive more or less traffic than it was engineered for, leading to suboptimal service performance or even stressing some locations while leaving others underutilized. Operators constantly face this scenario that

when expanding an anycast service. Operators cannot easily directly estimate future query distributions based on proposed anycast network engineering decisions.

To address this need and estimate the query loads based on changing, in particular expanding, anycast service changes [Vries17b] developed a new technique enabling operators to carry out active measurements, using an open-source tool called Verfploeter (available at [VerfSrc]). The results allow the creation of detailed anycast maps and catchment estimates. By running verfploeter combined with a published IPv4 "hit list", DNS can precisely calculate which remote prefixes will be matched to each anycast instance in a network. At the moment of this writing, Verfploeter still does not support IPv6 as the IPv4 hit lists used are generated via frequent large scale ICMP echo scans, which is not possible using IPv6.

As proof of concept, [Vries17b] documents how it verfploeter was used to predict both the catchment and query load distribution for a new anycast instance deployed for b.root-servers.net. Using two anycast test instances in Miami (MIA) and Los Angeles (LAX), an ICMP echo query was sent from an IP anycast addresses to each IPv4 /24 network routing block on the Internet.

The ICMP echo responses were recorded at both sites and analyzed and overlayed onto a graphical world map, resulting in an Internet scale catchment map. To calculate expected load once the production network was enabled, the quantity of traffic received by b.root-servers.net's single site at LAX was recorded based on a single day's traffic (2017-04-12, DITL datasets [Ditl17]). [Vries17b] predicted that 81.6% of the traffic load would remain at the LAX site. This estimate by verfploeter turned out to be very accurate; the actual measured traffic volume when production service at MIA was enabled was 81.4%.

Verfploeter can also be used to estimate traffic shifts based on other BGP route engineering techniques (for example, AS path prepending or BGP community use) in advance of operational deployment. [Vries17b] studied this using prepending with 1-3 hops at each instance and compared the results against real operational changes to validate the techniques accuracy.

An important operational takeaway [Vries17b] provides is how DNS operators can make informed engineering choices when changing DNS anycast network deployments by using Verfploeter in advance. Operators can identify sub-optimal routing situations in advance with significantly better coverage than using other active measurement platforms such as RIPE Atlas. To date, Verfploeter has been deployed

on a operational testbed (Anycast testbed) [AnyTest], on a large unnamed operator and is run daily at b.root-servers.net[Vries17b].

Operators are encouraged to use active measurement techniques like Verfploeter in advance of potential anycast network changes to accurately measure the benefits and potential issues ahead of time.

6. C4: When under stress, employ two strategies

DDoS attacks are becoming bigger, cheaper, and more frequent [Moura16b]. The most powerful recorded DDoS attack against DNS servers to date reached 1.2 Tbps by using IoT devices [Perlroth16]. How should a DNS operator engineer its anycast authoritative DNS server react to such a DDoS attack? [Moura16b] investigates this question using empirical observations grounded with theoretical option evaluations.

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 is overloaded with incoming traffic, operators have two options:

- They can withdraw its routes, pre-prepend its AS route to some or all of its neighbors, perform other traffic shifting tricks (such as reducing route announcement propagation using BGP communities[RFC1997]), or by communicating with its upstream network providers to apply filtering (potentially using FlowSpec [RFC5575]). These techniques shift both legitimate and attack traffic to other anycast instances (with hopefully greater capacity) or to block traffic entirely.

- Alternatively, operators can be become a degraded absorber by continuing to operate, knowing dropping incoming legitimate requests due to queue overflow. However, this approach will also

absorb attack traffic directed toward its catchment, hopefully protecting the other anycast instances.

[Moura16b] saw both of these behaviors deployed in practice by studying instance reachability and route-trip time (RTTs) in the DNS root events. When withdraw strategies were deployed, the stress of increased query loads were displaced from one instance to multiple other sites. In other observed events, one site was left to absorb the brunt of an attack leaving the other sites to remain relatively less affected.

Operators should consider having both a anycast site withdraw strategy and a absorption strategy ready to be used before a network overload occurs. Ideally, these should be encoded into operating playbooks with defined site measurement guidelines for which strategy to employ based on measured data from past events.

[Moura16b] speculates that careful, explicit, and automated management policies may provide stronger defenses to overload events. DNS operators should be ready to employ both traditional filtering approaches and other routing load balancing techniques (withdraw/prepend/communities or isolate instances), where the best choice depends on the specifics of the attack.

Note that this consideration refers to the operation of just one anycast service point, i.e., just one anycasted IP address block covering one NS record. However, DNS zones with multiple authoritative anycast servers may also expect loads to shift from one anycasted server to another, as resolvers switch from on authoritative service point to another 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 50 ms response to a new DNS query may be considered fast, but a less than 1 ms response to a cached entry is far faster. [Moura18b] showed that caching also protects users from short outages and even significant DDoS attacks.

DNS record TTLs (time-to-live values) [RFC1034][RFC1035] directly control cache durations and 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 has examined their interaction with DNS[Moura18b], but until [Moura19a] no research provided considerations about the benefits of various TTL value choices. To study this, Moura et. al. [Moura19a] carried out a measurement study investigating TTL choices and their impact on user

experiences in the wild. They performed this study independent of specific resolvers (and their caching architectures), vendors, or setups.

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

- As discussed, longer TTLs lead to a longer cache life, resulting in faster responses. [Moura19a] measured this in the wild and showed that by increasing the TTL for .uy TLD from 5 minutes (300s) to 1 day (86400s) the latency measured from 15k Atlas vantage points changed significantly: the median RTT decreased from 28.7ms to 8ms, and the 75%ile decreased from 183ms to 21ms.

- Longer caching times also results in lower DNS traffic: authoritative servers will experience less traffic with extended TTLs, as repeated queries are answered by resolver caches.

- Consequently, longer caching results in a lower overall cost if DNS is metered: some DNS-As-A-Service providers charge a per query (metered) cost (often in addition to a fixed monthly cost).

- Longer caching is more robust to DDoS attacks on DNS infrastructure. [Moura18b] also measured and show that DNS caching can greatly reduce the effects of a DDoS on DNS, provided that caches last longer than the attack.

- However, shorter caching supports deployments that may require rapid operational changes: An easy way to transition from an old server to a new one is to simply change the DNS records. Since there is no method to remotely remove cached DNS records, the TTL duration represents a necessary transition delay to fully shift from one server to another. Thus, low TTLs allow for more rapid transitions. However, when deployments are planned in advance (that is, longer than the TTL), it is possible to lower the TTLs just-before a major operational change and raise them again afterward.

- Shorter caching can also help with a DNS-based response to DDoS attacks. Specifically, some DDoS-scrubbing services use the DNS to redirect traffic during an attack. Since DDoS attacks arrive unannounced, DNS-based traffic redirection requires the TTL be kept quite low at all times to allow operators to suddenly have their zone served by a DDoS-scrubbing service.

- Shorter caching helps DNS-based load balancing. Many large services are known to rotate traffic among their servers using DNS-based load balancing. Each arriving DNS request provides an

opportunity to adjust service load by rotating IP address records (A and AAAA) to the lowest unused server. Shorter TTLs may be desired in these architectures to react more quickly to traffic dynamics. Many recursive resolvers, however, have minimum caching times of tens of seconds, placing a limit on this form of agility.

Given these considerations, the proper choice for a TTL depends in part on multiple external factors -- no single recommendation is appropriate for all scenarios. Organizations must weigh these trade-offs and find a good balance for their situation. Still, some guidelines can be reached when choosing TTLs:

- For general DNS zone owners, [Moura19a] recommends a longer TTL 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 and may, even, literally provide a cost savings.

- For registry operators: TLD and other public registration operators (for example most ccTLDs and .com, .net, .org) that host many delegations (NS records, DS records and "glue" records), [Moura19a] demonstrates that most resolvers will use the TTL values provided by the child delegations while the others some will choose the TTL provided by the parent's copy of the record. As such, [Moura19a] recommends longer TTLs (at least an hour or more) for registry operators as well for child NS and other records.

- Users of DNS-based load balancing or DDoS-prevention services may require shorter TTLs: TTLs may even need to be as short as 5 minutes, although 15 minutes may provide sufficient agility for many operators. There is always a tussle between shorter TTLs providing more agility against all the benefits listed above for using longer TTLs.

- Use of A/AAAA and NS records: The TTLs for A/AAAA records should be shorter to or equal to the TTL for the corresponding NS records for in-bailiwick authoritative DNS servers, since [Moura19a] finds that once an NS record expires, their associated A/AAAA will also be re-queried when glue is required to be sent by the parents. For out-of-bailiwick servers, A, AAAA and NS records are usually all cached independently, so different TTLs can be used effectively if desired. In either case, short A and AAAA records may still be desired if DDoS-mitigation services are required.

8. Security considerations

This document discusses applying measured research results to operational deployments. Most of the considerations affect mostly operational practice, though a few do have security related impacts.

Specifically, C4 discusses a couple of strategies to employ when a service is under stress from DDoS attacks and offers operators additional guidance when handling excess traffic.

Similarly, C5 identifies the trade-offs with respect to the operational and security benefits of 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 the client-to-resolver and resolver-to-authoritative cases.

10. IANA considerations

This document has no IANA actions.

11. Acknowledgements

This document is a summary of the main considerations of six research works performed by the authors and others. This document would not have been possible without the hard work of these authors and co-authors:

- 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 reviewers of this draft that offered valuable suggestions: 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 acknowledged in the papers this document summarizes for helping produce the results: 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>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[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>.

[Moura20a]
Moura, G., Heidemann, J., Hardaker, W., Bulten, J., Ceron, J., and C. Hesselman, "Old but Gold: Prospecting TCP to Engineer DNS Anycast (extended)", Technical Report ISI-TR-740 USC/Information Sciences Institute. , June 2020, <https://www.isi.edu/~johnh/PAPERS/Moura20a.pdf>.

[Moura20b]
Moura, G., Castro, S., Hardaker, W., Wullink, M., and C. Hesselman, "Clouding up the Internet: how centralized is DNS traffic becoming?", ACM 2020 Internet Measurement Conference, DOI 10.1145/3419394.3423625, October 2020, <http://giovane-moura.nl/resources/paper/Moura20b.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