DNSOP Working Group
Internet-Draft
Intended status: Informational
Expires: June 23, 2019

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

December 20, 2018

Recommendations for Authoritative Servers Operators
draft-moura-dnsop-authoritative-recommendations-01

Abstract

This document summarizes recent research work exploring DNS configurations and offers specific, tangible recommendations to operators for configuring authoritative servers.

This document is not an Internet Standards Track specification; it is published for informational purposes.

Ed note

Text inside square brackets ([RF:ABC]) refers to individual comments we have received about the draft, and enumerated under <https://github.com/gmmoura/draft-moura-dnsop-authoritative-recommendations/blob/master/reviews/reviews-dnsop.md>. They will be removed before publication.

This draft is being hosted on GitHub - <https://github.com/gmmoura/draft-moura-dnsop-authoritative-recommendations>, where the most recent version of the document and open issues can be found. The authors gratefully accept pull requests.

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 June 23, 2019.

Table of Contents

1. Introduction

The domain name system (DNS) has main two types of DNS servers: authoritative servers and recursive resolvers. Figure 1 shows their relationship. An authoritative server knows the content of a DNS zone from local knowledge, and thus can answer queries about that zone needing to query other servers [RFC2181]. A recursive resolver is a program that extracts information from name servers in response

to client requests [RFC1034]. A client, in Figure 1, is shown as stub, which is shorthand for stub resolver [RFC1034] that is typically located within the client software.

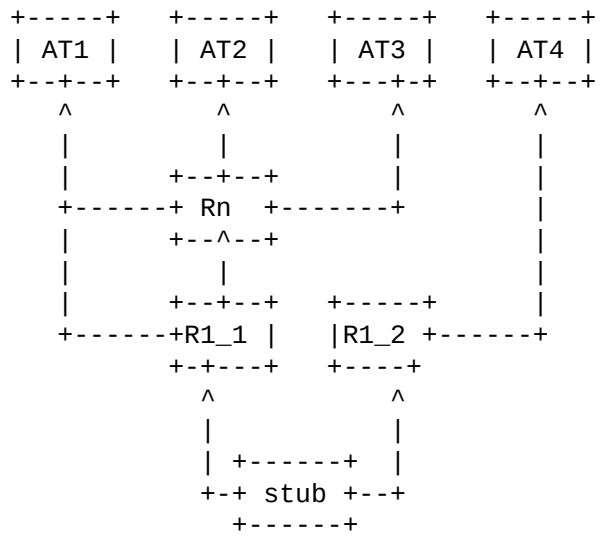

Figure 1: Relationship between recursive resolvers (R) and authoritative name servers (AT)

DNS queries contribute to user's 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 the deployment of multiple servers for the same zone [RFC1035] (AT1--AT4 in Figure 1), the use of IP anycast [RFC1546][RFC4786][RFC7094] and by using load balancers to support multiple servers inside a single (potentially anycasted) site. As a consequence, there are many possible ways a DNS provider can engineer its production authoritative server network, with multiple viable choices and no single optimal design.

This document summarizes recent research work exploring DNS configurations and offers specific tangible recommendations to DNS authoritative servers operators (DNS operators hereafter). [RF:JAb2]], [RF:MSJ1], [RF:DW2]. The recommendations (R1-R6) 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.

[RF:JAb1, Issue#2]. These recommendations are designed for operators of "large" authoritative servers for domains like TLDs. "Large" authoritative servers mean those with a significant global user population. These recommendations 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 recommendations 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.

2. R1: Use equaly strong IP anycast in every authoritative server to achieve even load distribution

Authoritative DNS servers operators announce their authoritative servers in the form of Name Server (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 site and allowing Internet routing (BGP[RFC4271]) to associate clients with their topologically nearest anycast site. 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 9,000 RIPE Atlas probes (Vantage Points--VPs) 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 tend 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 recursives 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 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 site in Europe will always have a high latency to resolvers in California, for example), [Mueller17b] recommends to DNS operators that they deploy equally strong IP anycast in every authoritative server (and, consequently, to phase out unicast), so they can deliver 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 recommendation was deployed at the ".nl" TLD zone, which originally had a mixed unicast/anycast setup; since early 2018 it now has 4 anycast authoritative name servers.

3. R2: 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 sites, i.e., the number of global locations from which the same address is announced with BGP. Intuitively, one could think that more sites 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 sites and the performance of a service (latency-wise, RTT) and measured the overall performance of four DNS Root servers, namely C, F, K and L, from more than 7.9K RIPE Atlas probes.

[Schmidt17a] found that C-Root, a smaller anycast deployment consisting of only 8 sites, provided a very similar overall performance than that of the much larger deployments of K and L, with 33 and 144 sites respectively. A median RTT was measured between 30ms and 32ms for C, K and L roots, and 25ms for F.

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

4. R3: Collecting Detailed Anycast Catchment Maps Ahead of Actual Deployment Can Improve Engineering Designs

An anycast DNS service may have several dozens or even hundreds sites (such as L-Root does). Anycast leverages Internet routing to distribute the incoming queries to a service’s distributed anycast sites; in theory, BGP (the Internet’s defacto routing protocol) forwards incoming queries to a nearby anycast site (in terms of BGP distance). However, usually queries are not evenly distributed across all anycast sites, as found in the case of L-Root [IcannHedge18].

Adding new sites to an anycast service may change the load distribution across all sites, leading to suboptimal usage of the service or even stressing some sites while others remain underutilized. This is a scenario that operators constantly face when expanding an anycast service. Besides, when setting up a new anycast service instance, operators cannot directly estimate the query distribution among the sites in advance of enabling the site.

To estimate the query loads across sites 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 site). To do that, [Vries17b] developed a new technique enabling operators to carry out active measurements, using aan 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 sites.

[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 sites in Miami (MIA) and Los Angeles (LAX) from the operational B-Root server, they sent ICMP echo packets to IP addresses from each IPv4 /24 in on the Internet using a source address within the anycast prefix. Then, they recorded which site 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 site. 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.

Besides that, Verfploeter can also be used to estimate how traffic shifts among sites 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 site. 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 sites 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 recommendation is therefore to deploy a small test Verfploeter-enabled platform in advance at a potential anycast site may reveal the realizable benefits of using that site as an anycast interest, potentially saving significant financial and labor costs of deploying hardware to a new site that was less effective than as had been hoped.

5. R4: 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 and sites. Ultimately, the relationship between the DNS provider's network and a client's ISP will determine which anycast site will answer for queries for a given client. As a consequence, when an anycast authoritative server is under attack, the load that each anycast site receives is likely to be unevenly distributed (a function of the source of the attacks), thus some sites 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 sites may have different capacity (bandwidth, CPU, etc.), making a decision about how to react to stress becomes even more difficult.

In practice, an anycast site 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, ([RF:Issue3]) 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 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 sites (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 sites.

[Moura16b] saw both of these behaviors in practice in the Root DNS events, observed through site 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 site 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 sites and the attackers. Some policies are explicit, such as the choice of local-only anycast sites, or operators removing a site 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, an area currently under study. For DNS operators, that means that besides traditional filtering, two other options are available (withdraw/prepend/communities or isolate sites), and the best choice depends on the specifics of the attack.

6. R5: Consider longer time-to-live values whenever possible

In a DNS response, each resource record is accompanied by a time-to-live value (TTL), which "describes how long a RR can be cached before it should be discarded" [RFC1034]. The TTL values are set by zone owners in their zone files - either specifically per record or by using default values for the entire zone. Sometimes the same resource record may have different TTL values - one from the parent and one from the child DNS server. In this cases, resolvers are expected to prioritize the answer according to Section 5.4.1 in [RFC2181].

While set by authoritative server operators (labeled "AT"s in Figure 1), the TTL value in fact influences the behavior of recursive resolvers (and their operators - "Rn" in the same figure), by setting an upper limit on how long a record should be cached before discarded. In this sense, caching can be seen as a sort of "ephemeral replication", i.e., the contents of an authoritative server are placed at a recursive resolver cache for a period of time up to the TTL value. Caching improves response times by avoiding repeated queries between recursive resolvers and authoritative.

Besides improving performance, it has been argued that caching plays a significant role in protecting users during DDoS attacks against authoritative servers. To investigate that, [Moura18b] evaluates the role of caching (and retries) in DNS resiliency to DDoS attacks. Two authoritative servers were configured for a newly registered domain

and a series of experiments were carried out using various TTL values (60,1800, 3600, 86400s) for records. Unique DNS queries were sent from roughly 15,000 vantage points, using RIPE Atlas.

[Moura18b] found that , under normal operations, caching works as expected 70% of the times in the wild. It is believe that complex recursive infrastructure (such as anycast recursives with fragmented cache), besides cache flushing and hierarchy explains these other 30% of the non-cached records. The results from the experiments were confirmed by analyzing authoritative traffic for the .nl TLD, which showed similar figures.

[Moura18b] also emulated DDoS attacks on authoritative servers were emulated by dropping all incoming packets for various TTLs values. For experiments when all authoritative servers were completely unreachable, they found that TTL value on the DNS records determined how long clients received responses, together with the status of the cache at the attack time. Given the TTL value decreases as time passes at the cache, it protected clients for up to its value in cache. Once the TTL expires, there was some evidence of some recursives serving stale content [I-D.ietf-dnsop-terminology-bis]. Serving stale is the only viable option when TTL values expire in recursive caches and authoritative servers became completely unavailable.

They also emulated partial-failure DDoS failures were also emulated (similar to Dyn 2016 [Perlroth16], by dropping packet at rates of 50-90%, for various TTL values. They found that:

- Caching was a key component in the success of queries. For example, with a 50% packet drop rate at the authoritatives, most clients eventually got an answer.

- Recursives retries was also a key part of resilience: when caching could not help (for a scenario with TTL of 60s, and time in between probing of 10 minutes), recursive servers kept retrying queries to authoritatives. With 90% packet drop on both authoritatives (with TTL of 60s), 27% of clients still got an answer due to retries, at the price of increased response times. However, this came with a price for authroritative servers: a 8.1 times increase in normal traffic during a 90% packet drop with TTL of 60s, as recursives attempt to resolve queries - thus effectively creating "friendly fire".

Altogether, these results help to explain why previous attacks against the Roots were not noticed by most users [Moura18b] and why other attacks (such as Dyn 2016 [Perlroth16]) had significant impact on users experience: records on the Root zone have TTL values ranging

from 1 to 6 days, while some of unreachable Dyn clients had TTL values ranging from 120 to 300s, which limit how long records ought to be cached.

Therefore, given the important role of the TTL on user's experience during a DDoS attack (and in reducing ''friendly fire''), it is recommended that DNS zone owners set their TTL values carefully, using reasonable TTL values (at least 1 hour) whenever possible, given its role in DNS resilience against DDoS attacks. However, the choice of the value depends on the specifics of each operator (CDNs are known for using TTL values in the range of few minutes). The drawback of setting larger TTL values is that changes on the authoritative system infrastructure (e.g.: adding a new authoritative server or changing IP address) will take at least as long as the TTL to propagate among clients.

7. R6: Shared Infrastructure Risks Collateral Damage During Attacks

Co-locating services, such as authoritative servers, creates some degree of shared risk, in that stress on one service may spill over into another, resulting in collateral damage. Collateral damage is a common side-effect of DDoS, and data centers and operators strive to minimize collateral damage through redundancy, overcapacity, and isolation.

This has been seen in practice during the DDoS attack against the Root DNS system in November 2015 [Moura16b]. In this study, it was shown that two services not directly targeted by the attack, namely D-Root and the .nl TLD, suffered collateral damage. These services showed reduced end-to-end performance (i.e., higher latency and reduced reachability) with timing consistent with the DDoS event, strongly suggesting a shared resource with original targets of the attack.

Another example of collateral damage was the 1.2 Tbps attack against Dyn, a major DNS provider on October 2017 [Perlroth16]. As a result, many of their customers, including Airbnb, HBO, Netflix, and Twitter experienced issues with clients failing to resolve their domains, since the servers partially shared the same infrastructure.

It is recommended, therefore, when choosing third-party DNS providers, operators should be aware of shared infrastructure risks. By sharing infrastructure, there is an increased attack surface.

8. Security considerations

- to be added

9. IANA considerations

This document has no IANA actions.

10. Acknowledgements

4 This document is a summary of the main lessons of the research works mentioned on each recommendation here provided. As such, each author of each paper has a clear contribution.

Here we mention the papers co-authors and thank them for their work: Ricardo de O Schmidt, Wouter B de Vries, Moritz Mueller, Lan Wei, Cristian Hesselman, Jan Harm Kuipers, Pieter-Tjerk de Boer and Aiko Pras.

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.

11. References

11.1. Normative References

[I-D.ietf-dnsop-terminology-bis]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-14 (work in progress), September 2018.

[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>.

[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>.

11.2. 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/>.

[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>.

[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>.

[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