Stream: Independent Submission
RFC: 9199
Category: Informational
Published: March 2022
ISSN: 2070-1721
Authors: G. Moura
SIDN Labs/TU Delft
W. Hardaker
USC/Information Sciences Institute
J. Heidemann
USC/Information Sciences Institute
M. Davids
SIDN Labs

RFC 9199
Considerations for Large Authoritative DNS Server Operators

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 considerations or advice to authoritative DNS server operators. Authoritative server operators may wish to follow these considerations to improve their DNS services.

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

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

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9199.

Copyright Notice

Table of Contents

1. Introduction

This document summarizes recent research that explored the deployed DNS configurations and offers derived, specific, tangible advice to DNS authoritative server operators (referred to as "DNS operators" hereafter). The considerations (C1-C6) presented in this document are backed by peer-reviewed research, 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, readers are pointed to the pertinent publications where additional details are presented.

These considerations are designed for operators of "large" authoritative DNS servers, which, in this context, are servers with a significant global user population, like top-level domain (TLD) operators, run by either a single operator or multiple operators. Typically, these networks are deployed on wide anycast networks [RFC1546] [AnyBest]. These considerations may not be appropriate for smaller domains, such as those used by an organization with users in one unicast network or in a single 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 discussing large-scale DNS authoritative services.

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

2. Background

The DNS has two main 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].

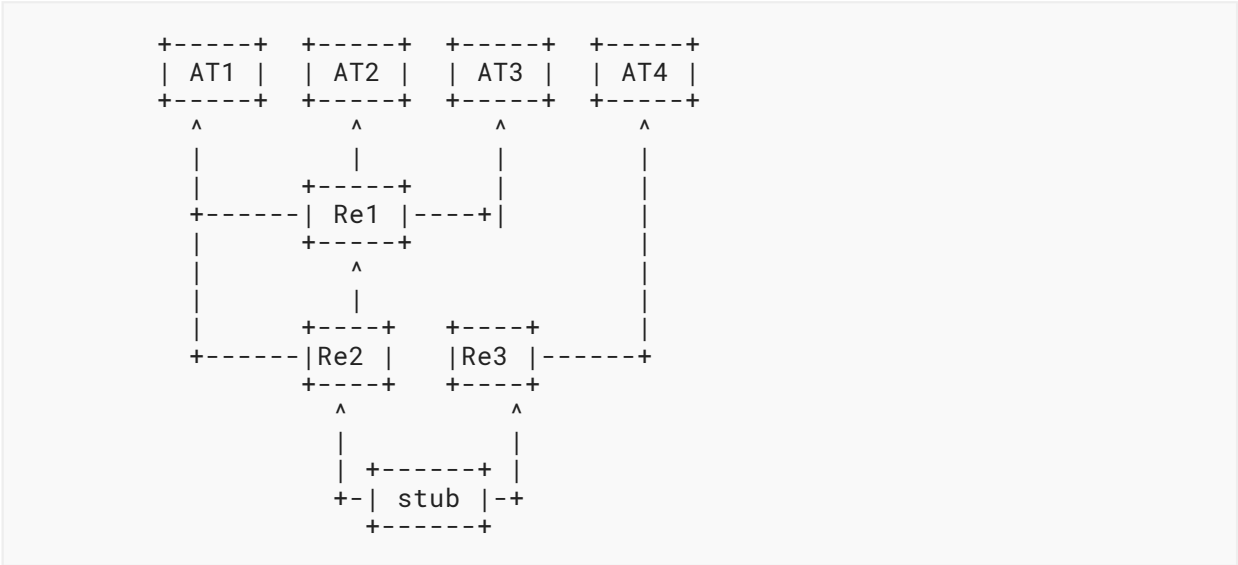

Figure 1: Relationship between Recursive Resolvers (Re) and Authoritative Name Servers (ATn)

DNS queries issued by a client contribute to a user's perceived latency and affect the user experience [Singla2014] 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 the 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 the following:

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 there is not necessarily a single optimal design.

3. Considerations

In the next sections, we cover the specific considerations (C1-C6) for conclusions drawn within academic papers about large authoritative DNS server operators. These considerations are conclusions reached from academic work that authoritative server operators may wish to consider in order to improve their DNS service. Each consideration offers different improvements that may impact service latency, routing, anycast deployment, and defensive strategies, for example.

3.1. C1: Deploy Anycast in Every Authoritative Server to Enhance Distribution and Latency

3.1.1. Research Background

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 (authoritative transfer (AXFR) [RFC5936] and incremental zone transfer (IXFR) [RFC1995]), coordinating the zone data they all return to their clients.

As discussed above, the DNS heavily relies upon replication to support high reliability, ensure capacity, and reduce latency [Moura16b]. The DNS has two complementary mechanisms for service replication: name server replication (multiple NS records) and anycast (multiple physical locations). Name server replication is strongly recommended for all zones (multiple NS records), and IP anycast is used by many larger zones such as the DNS root [AnyFRoot], most top-level domains [Moura16b], and many large commercial enterprises, governments, and other organizations.

Most DNS operators strive to reduce service latency for users, which is greatly affected by both of these replication techniques. However, because operators only have control over their authoritative servers and not over the client's recursive resolvers, it is difficult to ensure that recursives will be served by the closest authoritative server. Server selection is ultimately up to the recursive resolver's software implementation, and different vendors and even different releases employ different criteria to choose 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] describes the deployment of seven unicast authoritative name servers in different global locations and then queried them from more than 9000 Reseaux IP Europeens (RIPE) authoritative server operators and their respective recursive resolvers.

It was found in [Mueller17b] 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 they were not specific to any 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, it selects one of the alternatives.

3.1.2. Resulting Considerations

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 its usage of unicast and deploy a number of well-engineered 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.

The contribution of [Mueller17b] 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.

3.2. C2: Optimizing Routing is More Important than Location Count and Diversity

3.2.1. Research Background

When selecting an anycast DNS provider or setting up an anycast service, choosing the best number of anycast instances [RFC4786] [RFC7094] to deploy is a challenging problem. Selecting the right quantity and set of global locations that should send BGP announcements is tricky. Intuitively, one could naively think that more instances are better and that simply "more" will always lead to shorter response times.

This is not necessarily true, however. In fact, proper route engineering can matter more than the total number of locations, as found in [Schmidt17a]. To study the relationship between the number of anycast instances and the associated service performance, the authors measured the round-trip time (RTT) latency 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 (referred to as "C", "F", "K", and "L" hereafter) servers from more than 7,900 RIPE Atlas probes. RIPE Atlas is an Internet measurement platform with more than 12,000 global vantage points called "Atlas probes", and it is used regularly by both researchers and operators [RipeAtlas15a] [RipeAtlas19a].

In [Schmidt17a], the authors 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 RTTs for the C, K, and L root servers were all between 30-32 ms.

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 are more important to latency than the number of sites.

3.2.2. Resulting Considerations

The important conclusion from [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. Specifically, optimizing routing policies is more important than simply adding new instances. The authors 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 DoS attacks [Moura16b].

3.3. C3: Collect Anycast Catchment Maps to Improve Design

3.3.1. Research Background

An anycast DNS service may be deployed from anywhere and 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 nearest distributed anycast locations measured by the number of routing hops. However, queries are usually not evenly distributed across all anycast locations, as found in the case of L-Root when analyzed using Hedgehog [IcannHedgehog].

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 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 of an anycast service undergoing changes (in particular expanding), [Vries17b] describes the development of 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", the DNS can precisely calculate which remote prefixes will be matched to each anycast instance in a network. At the time 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 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 address to each IPv4 /24 network routing block on the Internet.

The ICMP echo responses were recorded at both sites and analyzed and overlaid 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, "day in the life" (DITL) datasets [Ditl17]). In [Vries17b], it was predicted that 81.6% of the traffic load would remain at the LAX site. This Verfploeter estimate 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, Autonomous System (AS) path prepending or BGP community use) in advance of operational deployment. This was studied in [Vries17b] using prepending with 1-3 hops at each instance, and the results were compared against real operational changes to validate the accuracy of the techniques.

3.3.2. Resulting Considerations

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 suboptimal routing situations in advance with significantly better coverage rather than using other active measurement platforms such as RIPE Atlas. To date, Verfploeter has been deployed on an operational testbed (anycast testbed) [AnyTest] on a large unnamed operator and is run daily at b.root-servers.net [Vries17b].

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

3.4. C4: Employ Two Strategies When under Stress

3.4.1. Research Background

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 Internet of Things (IoT) devices [Perlroth16]. How should a DNS operator engineer its anycast authoritative DNS server to 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 the BGP protocol maps clients to specific anycast instances using routing information. 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 when analyzing the root DNS events of November 2015 [Moura16b]. Given the fact that different instances may have different capacities (bandwidth, CPU, etc.), making a decision about how to react to stress becomes even more difficult.

In practice, when 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 communicate with its upstream network providers to apply filtering (potentially using FlowSpec [RFC8955] or the DDoS Open Threat Signaling (DOTS) protocol [RFC8811] [RFC9132] [RFC8783]). These techniques shift both legitimate and attack traffic to other anycast instances (with hopefully greater capacity) or block traffic entirely.
- Alternatively, operators can become degraded absorbers 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] describes seeing both of these behaviors deployed in practice when studying instance reachability and 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.

3.4.2. Resulting Considerations

Operators should consider having both an anycast site withdraw strategy and an absorption strategy ready to be used before a network overload occurs. Operators should be able to deploy one or both of these strategies rapidly. 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 common filtering approaches and other routing load-balancing techniques (such as withdrawing routes, prepending Autonomous Systems (ASes), adding communities, or isolating 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 one authoritative service point to another when attempting to resolve a name [Mueller17b].

3.5. C5: Consider Longer Time-to-Live Values Whenever Possible

3.5.1. Research Background

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 response of less than 1 ms to a cached entry is far faster. In [Moura18b], it was shown that caching also protects users from short outages and even significant DDoS attacks.

Time-to-live (TTL) values [RFC1034] [RFC1035] for DNS records directly control cache durations and affect latency, resilience, and the role of DNS in Content Delivery Network (CDN) server selection. Some early work modeled caches as a function of their TTLs [Jung03a], and recent work has examined cache interactions with DNS [Moura18b], but until [Moura19b], no research had provided considerations about the benefits of various TTL value choices. To study this, Moura et al. [Moura19b] 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:

- Longer TTLs, as discussed, lead to a longer cache life, resulting in faster responses. In [Moura19b], this was measured this in the wild, and it showed that by increasing the TTL for the .uy TLD from 5 minutes (300 s) to 1 day (86,400 s), the latency measured from 15,000 Atlas vantage points changed significantly: the median RTT decreased from 28.7 ms to 8 ms, and the 75th percentile decreased from 183 ms to 21 ms.
- Longer caching times also result in lower DNS traffic: authoritative servers will experience less traffic with extended TTLs, as repeated queries are answered by resolver caches.
- Longer caching consequently results in a lower overall cost if the DNS is metered: some providers that offer DNS as a Service 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. DNS caching was also measured in [Moura18b], and it showed that the effects of a DDoS on DNS can be greatly reduced, provided that the caches last longer than the attack.
- Shorter caching, however, 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 that 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 the 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.

3.5.2. Resulting Considerations

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, [Moura19b] recommends a longer TTL of at least one hour and ideally 4, 8, 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 cost savings.
- For TLD and other public registration operators (for example, most ccTLDs and .com, .net, and .org) that host many delegations (NS records, DS records, and "glue" records), [Moura19b] demonstrates that most resolvers will use the TTL values provided by the child delegations while some others will choose the TTL provided by the parent's copy of the record. As such, [Moura19b] 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 using shorter TTLs that provide more agility and using longer TTLs that include all the benefits listed above.
- Regarding the use of A/AAAA and NS records, the TTLs for A/AAAA records should be shorter than or equal to the TTL for the corresponding NS records for in-bailiwick authoritative DNS servers, since [Moura19b] finds that once an NS record expires, their associated A/AAAA will also be requeried 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.

3.6. C6: Consider the Difference in Parent and Children's TTL Values

3.6.1. Research Background

Multiple record types exist or are related between the parent of a zone and the child. At a minimum, NS records are supposed to be identical in the parent (but often are not), as are corresponding IP addresses in "glue" A/AAAA records that must exist for in-bailiwick authoritative servers. Additionally, if DNSSEC [RFC4033] [RFC4034] [RFC4035] [RFC4509] is deployed for a zone, the parent's DS record must cryptographically refer to a child's DNSKEY record.

Because some information exists in both the parent and a child, it is possible for the TTL values to differ between the parent's copy and the child's. [Moura19b] examines resolver behaviors when these values differed in the wild, as they frequently do -- often, parent zones have de facto TTL values that a child has no control over. For example, NS records for TLDs in the root zone are all set to 2 days (48 hours), but some TLDs have lower values within their published records (the TTLs for .cl's NS records from their authoritative servers is 1 hour). [Moura19b] also examines the differences in the TTLs between the NS records and the corresponding A/AAAA records for the addresses of a name server. RIPE Atlas nodes are used to determine what resolvers in the wild do with different information and whether the parent's TTL is used for cache lifetimes ("parent-centric") or the child's ("child-centric").

[Moura19b] found that roughly 90% of resolvers follow the child's view of the TTL, while 10% appear parent-centric. Additionally, it found that resolvers behave differently for cache lifetimes for in-bailiwick vs. out-of-bailiwick NS/A/AAAA TTL combinations. Specifically, when NS TTLs are shorter than the corresponding address records, most resolvers will requery for A/AAAA records for the in-bailiwick resolvers and switch to new address records even if the cache indicates the original A/AAAA records could be kept longer. On the other hand, the inverse is true for out-of-bailiwick resolvers: if the NS record expires first, resolvers will honor the original cache time of the name server's address.

3.6.2. Resulting Considerations

The important conclusion from this study is that operators cannot depend on their published TTL values alone -- the parent's values are also used for timing cache entries in the wild. Operators that are planning on infrastructure changes should assume that an older infrastructure must be left on and operational for at least the maximum of both the parent and child's TTLs.

4. 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 TTL values.

5. Privacy Considerations

This document does not add any new, practical 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 client-to-resolver and resolver-to-authoritative cases.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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>.

[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>.

[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>.

[RFC8783] Boucadair, M., Ed. and T. Reddy.K, Ed., "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", RFC 8783, DOI 10.17487/RFC8783, May 2020, <https://www.rfc-editor.org/info/rfc8783>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

[RFC9132] Boucadair, M., Ed., Shallow, J., and T. Reddy.K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", RFC 9132, DOI 10.17487/RFC9132, September 2021, <https://www.rfc-editor.org/info/rfc9132>.

7.2. Informative References

[AnyBest] Woodcock, B., "Best Practices in DNS Service-Provision Architecture", Version 1.2, March 2016, <https://meetings.icann.org/en/marrakech55/schedule/mon-tech/presentation-dns-service-provision-07mar16-en.pdf>.

[AnyFRoot] Woolf, S., "Anycasting f.root-servers.net", January 2003, <https://archive.nanog.org/meetings/nanog27/presentations/suzanne.pdf>.

[AnyTest] Tangled, "Tangled Anycast Testbed", <http://www.anycast-testbed.com/>.

[Ditl17] DNS-OARC, "2017 DITL Data", April 2017, <https://www.dns-oarc.net/oarc/data/ditl/2017>.

[IcannHedgehog] "hedgehog", commit b136eb0, May 2021, <https://github.com/dns-stats/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.C.M., Schmidt, R. de O., Heidemann, J., de Vries, W., Müller, M., Wei, L., and C. Hesselman, "Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event", ACM 2016 Internet Measurement Conference, DOI 10.1145/2987443.2987446, November 2016, <https://www.isi.edu/~johnh/PAPERS/Moura16b.pdf>.

[Moura18b] Moura, G.C.M., Heidemann, J., Müller, M., Schmidt, R. de O., 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>.

[Moura19b] Moura, G.C.M., Hardaker, W., Heidemann, J., and R. de O. Schmidt, "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/~hardaker/papers/2019-10-cache-me-ttls.pdf>.

[Mueller17b] Müller, M., Moura, G.C.M., Schmidt, R. de O., and J. Heidemann, "Recursives in the Wild: Engineering Authoritative DNS Servers", ACM 2017 Internet Measurement Conference, DOI 10.1145/3131365.3131366, November 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>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, DOI 10.17487/RFC4509, May 2006, <https://www.rfc-editor.org/info/rfc4509>.

[RFC8811] Mortensen, A., Ed., Reddy.K, T., Ed., Andreasen, F., Teague, N., and R. Compton, "DDoS Open Threat Signaling (DOTS) Architecture", RFC 8811, DOI 10.17487/RFC8811, August 2020, <https://www.rfc-editor.org/info/rfc8811>.

[RipeAtlas15a] RIPE Network Coordination Centre (RIPE NCC), "RIPE Atlas: A Global Internet Measurement Network", October 2015, <http://ipj.dreamhosters.com/wp-content/uploads/issues/2015/ipj18-3.pdf>.

[RipeAtlas19a] RIPE Network Coordination Centre (RIPE NCC), "RIPE Atlas", <https://atlas.ripe.net>.

[Schmidt17a] Schmidt, R. de O., Heidemann, J., and J. Kuipers, "Anycast Latency: How Many Sites Are Enough?", PAM 2017 Passive and Active Measurement Conference, DOI 10.1007/978-3-319-54328-4_14, March 2017, <https://www.isi.edu/%7ejohnh/PAPERS/Schmidt17a.pdf>.

[Singla2014] Singla, A., Chandrasekaran, B., Godfrey, P., and B. Maggs, "The Internet at the Speed of Light", 13th ACM Workshop on Hot Topics in Networks, DOI 10.1145/2670518.2673876, October 2014, <http://speedierweb.web.engr.illinois.edu/cspeed/papers/hotnets14.pdf>.

[VerfSrc] "Verfploeter Source Code", commit f4792dc, May 2019, <https://github.com/Woutifier/verfploeter>.

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

Acknowledgements

We would like to thank the reviewers of this document who offered valuable suggestions as well as comments at the IETF DNSOP session (IETF 104): Duane Wessels, Joe Abley, Toema Gavrichenkov, John Levine, Michael StJohns, Kristof Tuyteleers, Stefan Ubbink, Klaus Darilion, and Samir Jafferali.

Additionally, 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.

Contributors

This document is a summary of the main considerations of six research papers written by the authors and the following people who contributed substantially to the content and should be considered coauthors; this document would not have been possible without their hard work:

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

Authors' Addresses

Giovane C. M. Moura
SIDN Labs/TU Delft
Meander 501
6825 MD Arnhem
Netherlands
Phone: +31 26 352 5500
Email: giovane.moura@sidn.nl

Wes Hardaker
USC/Information Sciences Institute
PO Box 382
Davis, CA 95617-0382
United States of America
Phone: +1 (530) 404-0099
Email: ietf@hardakers.net

John Heidemann
USC/Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292-6695
United States of America
Phone: +1 (310) 448-8708
Email: johnh@isi.edu

Marco Davids
SIDN Labs
Meander 501
6825 MD Arnhem
Netherlands
Phone: +31 26 352 5500
Email: marco.davids@sidn.nl