IPv6-specific Misconfigurations in the DNS

Luuk Hendriks, Pieter-Tjerk de Boer and Aiko Pras
Design and Analysis of Communication Systems
University of Twente, the Netherlands
Email: {luuk.hendriks,p.t.deboer,a.pras}@utwente.nl

***Abstract*—With the Internet transitioning from IPv4 to IPv6, the number of IPv6-specific DNS records (AAAA) increases. Misconfigurations in these records often go unnoticed, as most systems are provided with connectivity over both IPv4 and IPv6, and automatically fall back to IPv4 in case of connection problems. With IPv6-only networks on the rise, such misconfigurations result in servers or services rendered unreachable.**

Using long-term active DNS measurements over multiple zones, we qualify and quantify these IPv6-specific misconfigurations. Applying pattern matching on AAAA records revealed which configuration mistakes occur most, the distribution of faulty records per DNS operator, and how these numbers evolved over time. We show that more than 97% of invalid records can be categorized into one of our ten defined main configuration mistakes. Furthermore, we show that while the number and ratio of invalid records decreased over the last two years, the number of DNS operators with at least one faulty AAAA record increased. This emphasizes the need for easily applicable checks in DNS management systems, for which we provide recommendations in the conclusions of this work.

I. Introduction

For years, use of IPv6 in the Internet has been increasing. This is often measured by big Internet companies analyzing the number of incoming connections from clients over IPv6. Examples of these types of figures are Google [1], Akamai [2], and Cisco 6lab [3]. Though these numbers mostly describe the eyeball-side of the Internet (though Cisco 6lab contains numbers on core networks too), these connections from the eyeballs are only possible with the server- and content-side providing IPv6 connectivity, and thus, having the necessary information available in the DNS. For IPv6, this means *AAAA* records should be present next to a possible A record. Not only does this provide access to clients without IPv6 connectivity, it also allows for a –possibly unnoticeable– fallback mechanism: client-side software often prefers and attempts to set up the connection over IPv6, but will switch back to IPv4 in case of connection failure. A very user-friendly scenario from the client perspective, though misconfigurations or problems might go unnoticed on the operator side.

Theoretically, these 'hidden' misconfigurations in the DNS would only become problematic if the Internet sees a sudden switch, becoming completely IPv6-only. Although a complete sudden switch is unrealistic, significant parts of the Internet are already deployed nowadays without any IPv4. Specifically mobile operators choose to only provide IPv6-connectivity towards their customers, and perform translation to IPv4 on the ISP-side whenever necessary. Effectively, misconfigured AAAA records in the DNS impair the end-user experience directly, as there is no fallback on the client-side anymore.

While DNS for IPv6 is not fundamentally different from IPv4, there are plenty of caveats for operators. For example, every IPv6-enabled network interface has a link-local address, *i.e.* an address in the `fe80::/64` range, which by nature is not globally routable. Ergo, such addresses have no place in the DNS. Another possible misconfiguration is the use of the so-called *IPv4-mapped IPv6 address*, *e.g.* `::ffff:192.0.2.1`. This type of address allows representation of an IPv4 address as an IPv6 address. Again, this is not a routable address, and has no place in the DNS.

With the increase in IPv6-only network deployments, finding and fixing misconfigurations in the DNS gains importance. In this study, we answer the following questions: 1) What share of AAAA records contain unrouteable IPv6 addresses? 2) What kinds of misconfiguration are apparent in the DNS, and how do they evolve in numbers over time? 3) How are these faulty records distributed in terms of DNS operators? 4) What can operators do to check for and prevent faulty AAAA records in their zones?

To answer these questions, we analyze two years worth of AAAA records spanning multiple Top Level Domains (TLDs) in the OpenINTEL [4] dataset. We distinguish faulty AAAA records and obtain related NS records. We classify different types of misconfigurations in these faulty AAAA records, and use the valid AAAA records to determine the ratio of valid to invalid AAAA records over time. With the NS records, the analysis per DNS operator is performed.

Contributions: In our analysis, we categorize ten different misconfigurations observed in AAAA records, and show that the most common misconfiguration is the use of IPv4-mapped IPv6 addresses. Furthermore, we show that while a small number of operators is responsible for a large share of misconfigurations, a large number of operators has a small number of misconfigurations. Finally, we propose a list of regular expressions that, combined, cover more than 95% of the observed misconfigured AAAA records.

II. Related Work

Several academic studies with a focus on misconfiguration or misbehavior in the DNS have been conducted. In [5], Pappas *et al.* focussed on different types of DNS misconfiguration (lame delegation, diminished server redundancy, and cyclic zone dependency) and concluded systematic checks are crucial in the DNS. Kazato *et al.* [6] classified erroneous DNS

978-3-901882-98-2

queries that generate erroneous answers, *e.g.* ServFails, based on captured traffic. Lu *et al.* [7] focus on the DNS in China, in 2014, and classify multiple types of misconfiguration in NS, MX and A records, but assess no AAAA or anything IPv6 related in their work.

From within the operator community, multiple studies, documents and projects touch the subjects of IPv6 and DNS. The following directly substantiate or motivate our work in this paper. Informational RFC4472, titled *Operational Considerations and Issues with IPv6 DNS* [8] mentions that limited scope addresses should never be published in the DNS: this concerns link-local and Unique Local Addresses (ULA). Other types of misconfiguration with regards to AAAA record content are not described. RFC 4291 [9], describes the IPv6 address architecture for all different types and scopes. It states IPv4-Compatible IPv6 Addresses (*e.g.* `::10.0.0.1`, thus without the leading `ffff` hextet) are deprecated. Dan Wing performs active measurements [10] to produce statistics on AAAA record availability and IPv6 connectivity for domains in the Alexa ranking. The statistics include numbers on IPv4-mapped and loopback addresses. Sander Steffann and Jan Žorž presented [11] an online tool to determine readiness of websites in DNS64/NAT64 scenarios, naturally revealing misconfigured AAAA records.

III. Methodology

The data we analyze is obtained from the OpenINTEL database. This database contains active measurements, conducted on a daily basis, since 2015. We focus on the contents of AAAA records, and, to answer our research questions concerning operators, we aggregate domains based on contents of the related NS records.

A. Querying the OpenINTEL database

Obtaining invalid AAAA records is done based on regular expressions in the SQL `SELECT` statement. We define *invalid* as anything not being in `2000::/3`, i.e. the Global Unicast range. Additionally, anything in the documentation range `2001:db8/32` is treated as invalid. These ranges translate into the following regular expressions,[1] respectively:

```
^[23][0-9a-f]{3}:
^2001:0?db8:
```

B. Classification of IPv6 addresses in the AAAA records

With the invalid AAAA records retrieved from the database, each record is tested for a match in the defined classes of misconfiguration. An overview of all classes and their respective regular expression is listed in Table I. Note that the last two classes in this table are not only determined by a regular expression. Firstly, the *::something* class overlaps with the *mapped-v4* and *mapped-v4-depr* classes. Therefore, if and only if an invalid AAAA record is not classified as being any of these two latter classes, though it does start with `::`, it is marked as *::something*. Secondly, the *UNKNOWN* class is simply everything that was not matched by any of the other regular expressions.

[1]Notation used in this paper is compliant with the POSIX Extended Regular Expressions notation, and expressed case-insensitively.

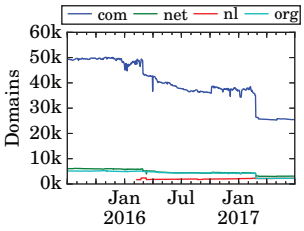

Fig. 1: Domains with invalid AAAA records over time.

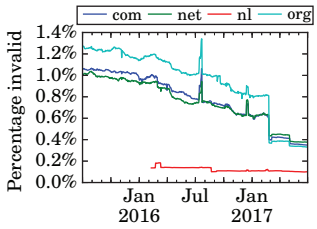

Fig. 2: Percentage of invalid AAAA records over time.

C. Defining DNS operators from NS records

In order to reason about the data on a per-operator level, a third query is performed, based on the invalid AAAA records. For every invalid AAAA record, we lookup the NS record for that domain. From that NS record, the TLD and Second Level Domain (SLD) are extracted, e.g.:

`ns1.example.com` results in `example.com`

Thus, we define `example.com` as the DNS operator for this domain.

IV. Results

We present our results analogously to and in order of the research questions presented in Section I. Additionally, conspicuous peaks and troughs in graphs are discussed on a case-by-case basis.

A. Invalid AAAA records in the DNS over time

Visualized in Figure 1, we see decreasing numbers of invalid AAAA records over the two year timespan of the data. This is a positive development, but to put these numbers in perspective, one needs the total number of AAAA records in the DNS. In Figure 2, the share of invalid records is visualized, calculated as shown in Expression 1.

$$\frac{\text{No. of invalid records}}{\text{No. of invalid + No. of valid records}} \quad (1)$$

The graph clearly shows decreasing relative numbers as well, with similar developments across .com, .net and .org. The .nl zone shows less decrease, though features a significantly better ratio. At the end of the analyzed time window, all zones have less than 0.5% invalid AAAA records. At 0.4% for .com, .net and .org, it means still one in 250 domains is unreachable in an IPv6-only environment.

1) Relative peaks, July/December 2016: Comparing the graphs in Figure 1 and Figure 2, we see clear peaks in the latter, half-way July 2016 and half-way December 2016, while there is no peak in the absolute numbers. This difference hints at a decrease of valid AAAA records across multiple zones: this is indeed the case, as we found a drop of more than 20M valid AAAA records in July, spread over .com, .net and org. Again in December, around 10M valid records disappeared from the DNS.

TABLE I: Matching and examples for the defined classes of misconfiguration

class	Regular Expression	Example	Description
mapped-v4	`^::ffff:([0-9]+\.){3}[0-9]+$`	`::ffff:1.2.3.4`	IPv6-mapped IPv4 address
unspec	`^::$`	`::`	The unspecified address, equivalent to 0.0.0.0 in IPv4
v4-hex::	`^[0-9a-f]{3,4}:[0-9a-f]{3,4}::$`	`c000:0201::`	IPv4 address in hexadecimal notation, appended with `::`
link-local	`^fe80:`	`fe80::a:b:c:d`	Link-local range, `fe80::/64`
localhost	`^::1$`	`::1`	Localhost address, equivalent to 127.0.0.1 in IPv4
mapped-v4-depr	`^::([0-9]+\.){3}[0-9]+$`	`::1.2.3.4`	Deprecated IPv6-mapped IPv4 address
documentation	`^2001:0?db8:`	`2001:db8::1`	Documentation range, `2001:db8::/32`
ula	`^f[cd]..:`	`fc05::20:1`	Unique Local Address range, `fc00::/7`
well-known	`^64:ff9b:`	`64:ff9b::1.2.3.4`	The Well Known range, `64:ff9b::/96`, used in NAT64
multicast	`^ff0.:`	`ff02::16`	Multicast range, `ff00::/8`, restricted to well known prefixes*
::something	`^::`	`::`	Anything starting with `::`, not classified as *mapped-v4*(*-depr*)
UNKNOWN			Anything not classified by any of the above

* https://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml

2) Significant drop, February 2017: Visible in .net, .org, but most ostensibly in .com, is the drop at the end February, in 2017. We found this drop to be caused by a single operator, namely Directnic, retracting a large number of invalid AAAA records. More details about this change are provided in Section IV-B1.

B. Classification of misconfiguration

Applying pattern matching as specified with the regular expressions in Table I categorizes the invalid AAAA records into 12 distinct classes. The absolute number of records per class over time is depicted in Figure 3. The remainder after the classification, *i.e.* the records that do not match any of the defined patterns, is marked *UNKNOWN* and plotted as well.

Clearly, the majority of misconfigured domains have AAAA records that contain a *IPv6-mapped IPv4* address, labeled *mapped-v4*: more than 80% in 2015, and still 63% in 2017. While being the largest category of configuration mistakes, it is also seemingly the only type of misconfiguration that sees significant improvement, although there was a very substantial decrease for *v4-hex::* as well.

We now detail the most ostensible jumps in the graph:

1) Drop in mapped-v4*, February 2017:* As mentioned, this drop is caused by doings of or at Directnic. Nearly 12 000 domains that contained (the same) IPv6-mapped IPv4 address the day before, disappeared from the DNS. Both this particular and other IPv6-mapped addresses still occurred in other domains from this operator.

2) Absence of v4-hex::*, March - October 2016:* Again, this concerns only a single value, namely `da1e:2353::`, and disappears from multiple Top Level zones. Translating the bytes to IPv4 octets results in `218.30.35.83`, belonging to Chinanet. The NS records seemingly reveal just two operators, *idc1.cn* and *72dns.com*, with the same (IPv4) address, thus actually hinting at a single operator. Investigating the bump back up in October, we find the exact same address is being reintroduced, by the exact same operator(s).

3) Decrease in unspec*, March/April 2016:* On March 31 of 2016, 6000 domains from operator *listingdomains.com* containing the *unspecified* address disappear for a single day. The related AAAA records feature high similarity, namely real estate terminology, hinting at a single customer making changes in bulk.

TABLE II: Overview of categorized AAAA misconfigurations at the beginning and the end of the analyzed time window. Sorted by number of occurrences observed in July 2015.

class	July 2015	June 2017
mapped-v4	105981.0 (80.3%)	30711 (62.9%)
unspec	11655.0 (8.8%)	8435 (17.3%)
v4-hex::	7346.0 (5.6%)	30 (0.1%)
link-local	2744.0 (2.1%)	4947 (10.1%)
localhost	2434.0 (1.8%)	2420 (5.0%)
UNKNOWN	767.0 (0.6%)	1189 (2.4%)
mapped-v4-depr	466.0 (0.4%)	533 (1.1%)
documentation	168.0 (0.1%)	140 (0.3%)
::something	155.0 (0.1%)	54 (0.1%)
ula	152.0 (0.1%)	195 (0.4%)
well-known	70.0 (0.1%)	122 (0.2%)
multicast	18.0 (0.0%)	80 (0.2%)
mapped-v4-hex	nan (nan%)	1 (0.0%)

C. Distribution of misconfiguration over DNS operators

We found that a small number of operators is accountable for a large share of the misconfigurations in the DNS. Looking at the distribution of faulty records per operator, shown in Figure 4, we find 80% is caused by a handful of operators, as marked by the red box on the tail of the distribution. At the same time, more than 90% of operators have less than 10 invalid AAAA records (in our dataset). The number of unique operators that have one or more misconfigured records, is increasing. Figure 5 shows a clear increase of roughly 30% over the two-year timespan of our data. Assessing the different types of misconfiguration, we found that almost 95% of operators only have misconfigurations of one or two types in their zones. For the actual content of the AAAA records, 80% show one and the same single value. These numbers hint at the use of either (invalid) default values, automation, or simple copy/paste errors by operators.

D. Other findings

Besides the answers to the research questions, we discovered several other things while working with the data.

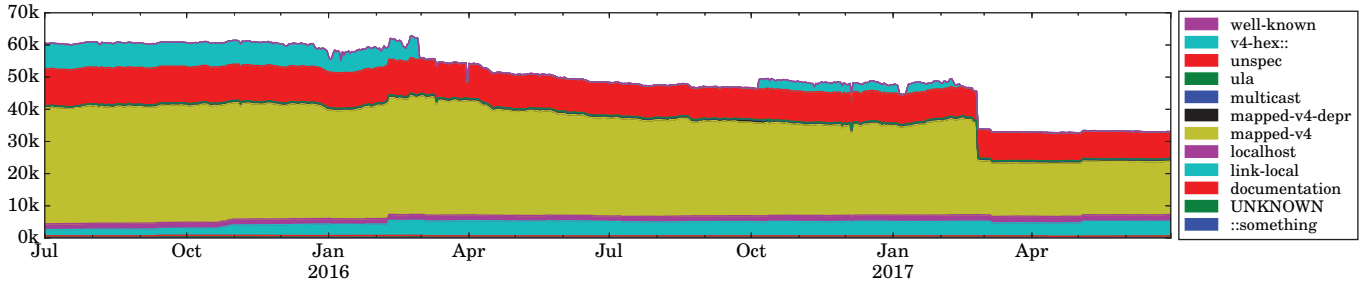

Fig. 3: Number of faulty AAAA records per class over time. All zones combined.

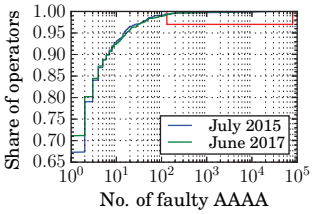

Fig. 4: Cumulative distribution of number of invalid AAAA records per second level domain NS.

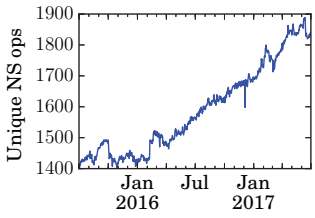

Fig. 5: Unique NS operators with at least one faulty AAAA record, over time, all zones.

For 300 domains in .com, records containing the IPv6 localhost address featured IPv4 counterparts also containing the (IPv4) localhost adddress. This means that at least some misconfigurations are not IPv6 specific per se.

Furthermore, in attempts to drill down the *UNKNOWN* class, we found a number of addresses that seem to miss the first hexadecimal character but were valid otherwise. Adding the first character (which can only be either a 2 or a 3 for valid Global Unicast addresses) allows a simple ping test. Out of 99 unique addresses, 37 responded, associated with 189 domains in total. Prominent in this set were 96 domains which were all related to a single organization, again hinting at a copy/paste mistake during configuration.

V. Discussion

IPv6 addresses can be represented in multiple textual formats. This flexibility introduces caveats when applying pattern matching. For example, a 16-bit field may be represented with leading zeroes: `2001:db8::2:1` is equivalent to `2001:db8::0002:0001`. The data we used in our study contains IPv6 addresses in string format, produced by the `inet_ntop` system call. Therefore, we could apply the regular expressions as presented in this paper. Implementing these exact regular expressions to check user input in a form might not suffice per se, as the user might use unforeseen (leading) zeroes. So, normalizing the input prior to the pattern matching is recommended. However, aiming at detecting unroutable addresses, the main regular expression to match anything outside `2000::/3` is not prone to alternative representations, as long as it is performed case insensitively. More information on different textual representations of IPv6 addresses can be found in [12].

VI. Conclusions

Having analyzed the share of AAAA records containing unroutable IPv6 addresses, we find both the absolute number as well as the ratio to valid AAAA records decreasing. In July 2015, the ratio of invalid AAAA records to all AAAA records was 1% in .com and .net, and even 1.25% in .org. In absolute numbers, the initial number of 60 000 domains containing invalid AAAA records (cumulative over all zones) in July 2015 decreased to roughly 30 000 in June 2017. At that point in time in 2017, the measurements for .nl were also included, making the drop of faulty configured domains even more significant.

The most frequent misconfiguration was and is the IPv6-mapped IPv4 address. Even though it is the misconfiguration that saw the most significant improvement, it still accounts for more than 60% of invalid AAAA records. Though overall numbers are improving, the number of individual DNS operators with misconfigured AAAA records is increasing, while the share of top offenders decreased. Still, the top 1% of operators shows 100 or more misconfigured AAAA records, and the top 0.1% is responsible for 1000 or more. The majority of operators show at most two different types of misconfigurations, and few different values in the records, strongly hinting at automated processes in their DNS management.

With the two regular expressions presented in the methodology, operators can find and prevent unroutable addresses in their (customers') records. With the regular expressions used in the classification (covering 97% of the invalid records in our dataset), more specific info can be provided to end-users and customers. Only matching on the *mapped-v4*, *unspecified* and *link-local* classes already covers 90%, thus could greatly reduce and prevent misconfigurations with minimum effort.

With still roughly 0.4% of all AAAA records being invalid and thus breaking connectivity on IPv6-only networks, there are substantial gains to be made, beneficial to both end-users and content/service providers.

References

[1] Google, "IPv6 Statistics," https://www.google.com/intl/en/ipv6/statistics.html, 2017.

[2] Akamai, "IPv6 Adoption Visualization," https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/state-of-the-internet-ipv6-adoption-visualization.jsp, 2017.
[3] Cisco, "6lab," http://6lab.cisco.com/stats/, 2017.
[4] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 34, no. 6, pp. 1877–1888, 2016.
[5] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang, "Impact of Configuration Errors on DNS Robustness," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 319–330, Aug. 2004.
[6] Y. Kazato, K. Fukuda, and T. Sugawara, "Towards Classification of DNS Erroneous Queries," in *Proceedings of the 9th Asian Internet Engineering Conference*, ser. AINTEC '13. New York, NY, USA: ACM, 2013, pp. 25–32.
[7] K. Lu, K. Dong, C. Wang, and H. Xu, "DNS configuration detection model," in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, Nov 2014, pp. 613–618.
[8] A. Durand, J. Ihren, and P. Savola, "Operational Considerations and Issues with IPv6 DNS," Internet Requests for Comments, RFC Editor, RFC 4472, April 2006.
[9] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," Internet Requests for Comments, RFC Editor, RFC 4291, February 2006, http://www.rfc-editor.org/rfc/rfc4291.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4291.txt
[10] D. Wing, "AAAA and IPv6 Connectivity Statistics," http://www.employees.org/~dwing/aaaa-stats/, 2017.
[11] J. Žorž and S. Steffann, "NAT64 Experiments," https://ripe74.ripe.net/presentations/133-Jan_Zorz-NAT64-Check-v3.4.pdf, 2017, presentation at RIPE74, Budapest.
[12] S. Kawamura and M. Kawashima, "A recommendation for ipv6 address text representation," Internet Requests for Comments, RFC Editor, RFC 5952, August 2010.