Measuring Dual Stack: Uncovering Hidden Problems in Plain Sight (abstract)

Florian Streibelt
Max Planck Institute for Informatics
fstreibelt@mpi-inf.mpg.de

Oliver Gasser
Max Planck Institute for Informatics
oliver.gasser@mpi-inf.mpg.de

1 MOTIVATION

The Internet today not only consists of networks of networks, with IPv4 and IPv6 we are basically running two separate but intertwined Internets. And although it might never happen, we have to ask: How do we prepare for turning off IPv4 for good? And if we do so, how can we manage a transition without outages? What would break, if we turn off IPv4 now?

A lot of IPv6 related publications and websites focus on reporting metrics for IPv6 deployment or readyness, but they all only provide a partial view.

Examples are: Checking the ratio of IPv6 to IPv4 traffic[2], often using datasets not publicly available. Analyzing BGP data[1, 4], that only answers questions on the reachability of networks, and finally checking popular websites[5] if their webservers can be contacted via IPv6.

To answer our initial question, we are interested in checking the impact on end-users using the Internet without IPv4, so checking the availability of websites via IPv6 makes sense.

But what does IPv6-enabled actually mean looking at websites? We use the following definition:

> A website is (fully) IPv6-enabled, if a host that has only IPv6 connectivity to the Internet and runs a local resolver, is able to access a website and all embedded content hosted at the involved webservers.

Existing work[3] shows, that some websites are indeed experiencing problems when accessed via IPv6.

But why is it sometimes hard to tell if a website is IPv6-enabled? And how is answering this question related to service monitoring and outage detection in a dual stackdeployment?

The short answer is: because we are used to building resilient and self-healing systems, that try to hide error conditions as much as possible.

2 WHAT IS *FULL* IPV6 SUPPORT

Websites on today's Internet easily consist of more than 100 objects, requested from a multitude of different webservers. We consider a website *fully* IPv6-enabled only if they all can be fetched using IPv6 only.

To decide if this is the case, we have to check a) if each FQDN referenced in an objects URL is resolving to `AAAA` records, b) if the records are syntactically correct, c) if the correct content is returned via HTTP. And of course use IPv6 for all communication.

In order for this to work in an IPv6-*only* setting, the full DNS delegation chain from the root servers to the authoritative nameserver(s) of the domain in question need to be IPv6-enabled!

We thus note, that running a simple command like `dig +short AAAA www.example.org` can not be considered a valid methodology, to check if a website is IPv6-enabled, even when repeating this step for all referenced ressources.

The widespread use of CNAMEs introduces yet another pitfall when trying to assess IPv6 readyness. A `CNAME` in a zone served by a fully IPv6-enabled domain, can point to a name in a zone, where none of the nameservers is IPv6-enabled, leading to a situation comparable to lame delegation, where a nameserver is referred to, that is not available.

3 CHALLENGES FOR MEASURING

The Internet is built as a robust and resilient system, and today development is more and more focused on making it more accessible. To achieve that goal, information hiding is a valid approach. With Happy Eyeballs [6] a mechanism was introduced, that was specifically designed to operate in a 'partially broken' world, where IPv6 would not always be available. While in the DNS world we are used to iterating through all available nameservers of a domain, here we are switching protocols automatically. In contrast DNS switches to TCP only when the TC flag is set and using UDP is mandatory - for now.

But when implementing service monitoring or conducting measurements of misconfiguration, we need to keep these layers in mind.

This is also true when repurposing libraries built for writing (error tolerant) applications.

In a recent discussion we had with the maintainers of a certain DNS library for python, the maintainers explained that they are trying to build a library, that always tries to be RFC-compliant. Meaning that for example when receiving an invalid upstream response in some corner cases the received values, e.g., for a TTL, will be fixed. Or preventing non-RFC-compliant queries to be sent.

4 MEASURING THE IPV6-ONLY INTERNET

We are planning to conduct a measurement study on the 'true' IPv6-readyness of websites, analyzing content availability of the embedded objects and specifically analyze the situation in the DNS. With that we hope to help operators identify missing links in their IPv6-enabled services and help the research community further identify common pitfalls.

For this we are currently implementing a measurement tool that acts like a recursive resolver. But in contrast it will be sending DNS queries to all available authoritative nameservers of a zone, and its parents, recording all error conditions. Additional challenges are avoiding unneccessary queries by implementing caching and rate limiting all queries to prevent getting blacklisted.

While working on the implementaion, we already experienced various, to us, unexpected behaviour in DNS libraries and observed strange responses from authoritative nameservers.

REFERENCES

[1] APNIC: IPv6 Capable Rate by country. https://stats.labs.apnic.net/ipv6.

[2] Google IPv6 Statistics. https://www.google.com/intl/en/ipv6/statistics.html.

[3] Steffie Jacob Eravuchira, Vaibhav Bajpai, Jürgen Schönwälder, and Sam Crawford. Measuring web similarity from dual-stacked hosts. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 181–187, 2016.

[4] Geoff Huston. IPv6 CIDR REPORT. https://www.cidr-report.org/v6/as2.0/.

[5] Eric Vyncke. IPv6 Deployment Aggregated Status. https://www.vyncke.org/ipv6status/.

[6] D. Wing and A. Yourtchenko. Happy eyeballs: Success with dual-stack hosts, April 2012. rfc6555.