Using Global Measurements Infrastructure to Understand the Evolution of the Internet

Vaibhav Bajpai and Jürgen Schönwälder*

Computer Science, Jacobs University Bremen, Germany
{v.bajpai,j.schoenwaelder}@jacobs-university.de

Abstract. The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be written using the *abstract* environment.

Keywords: Measurements, Management, IPv6

1 Introduction

2 Related Work

3 Preliminary Results

3.1 Measurements

A dual-stacked user when attempting to connect to a dual-stacked service traditionally prefers connecting over IPv6. This is because in POSIX systems, the internal domain name resolution system call `getaddrinfo(...)` [5] returns the list of addresses in an order that prioritizes an IPv6-upgrade path [7]. The dictated order can dramatically reduce the application responsiveness in situations where IPv6 connectivity is broken. This is because, the attempt to connect over an IPv4 address will take place only when the IPv6 connection attempt has timed out, which can be in the order of seconds.

This noticeable degraded user experience can be subverted by making applications apply the happy eyeballs algorithm [8]. The algorithm recommends that a dual-stacked application try resolving a dual-stacked service for both IPv4 and IPv6 addresses at once. If the resolver returns both addresses, the application must try a TCP `connect(...)` to both the resolved addresses and pick the one that completes first.

In this pursuit, to determine whether applications will use IPv4 or IPv6 on a dual stacked service, we developed `happy`, a simple TCP happy eyeballs probing tool. It uses non-blocking `connect(...)` calls to establish concurrent connections to a number of possible endpoints of a service. The tool, however, does not check

* This work was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) grant no. 317647 (Leone)

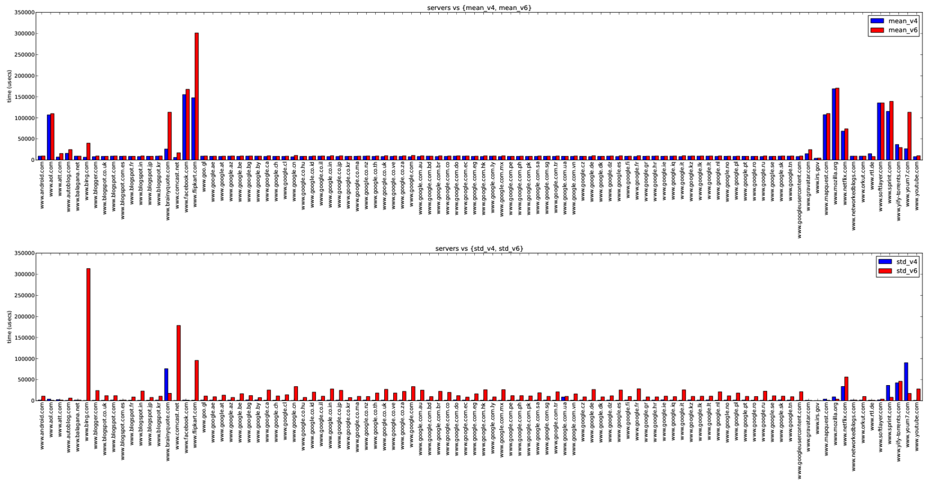

Fig. 1. servers vs `{mean, std}`

whether the endpoints of a given target all provide the same service. Hence, it is possible to impact the results by setting up fake servers that do not provide the service tested and which are designed and deployed with the only purpose to provide fast connection setup times.

We have cross-compiled `happy` for the OpenWRT [4] platform. As a result, the tool can now be run on widely deployed SamKnows probes [1], and the collected measurement data can be further analysed. In order to ascertain the value in this exercise, we prepared an internal test-bed of multiple measurement points. The measurement points have different flavors of IPv4 and IPv6 connectivity ranging from native IPv4, native IPv6, IPv6 tunnel broker endpoints [2], Teredo [6] and tunnelled IPv4. We used the top 100 domains compiled by Hurricane Electric Internet Services [2] and ran `happy` on the set of dual-stack services represented by these domains. A preliminary result comparing the time (mean and standard deviation) to make a TCP connection to each of these services over IPv4 and IPv6 is shown in Fig. 1. The measurement point represented in this plot is located at Braunschweig and has a native IPv4 and a IPv6 connection through the German Research Network [3]. The initial results show higher time variances when connections are made over IPv6. In addition, some of the related (and few of the unrelated) services show similar mean and standard deviations. These services either resolve to the same endpoint or a set of endpoints that belong to an allocated block. Digging through the `whois` information for each of the endpoints from their Regional Internet Registry (RIR) seems to indicate that

[1] `http://www.samknows.com`
[2] `http://bgp.he.net/ipv6-progress-report.cgi`
[3] `http://www.dfn.de`

major portion of the services map to a cloud of an address block owned by popular organizations like Google and Akamai Technologies as shown in Fig. 2.

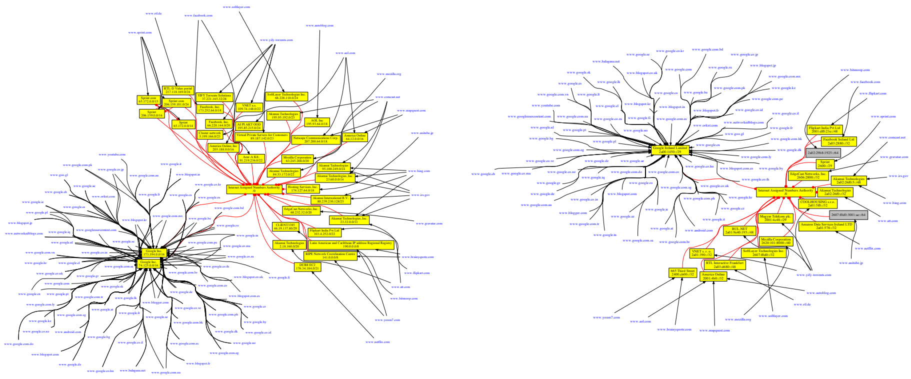

Fig. 2. IPv4 and IPv6 aggregation cloud

3.2 Local Management

As much as it is important to define and implement new tests on these measurement infrastructure, it is also equally pertinent to not only be able to install, update and delete these tests but also configure the entire suite of probes using a standardized protocol over the network. The Network Configuration (NETCONF) protocol [3] is particularly designed to cater to this problem. Towards this end, we have built a NETCONF server for the OpenWRT platform using the `libnetconf` [4] library and tested the implementation using our NETCONF Python API `ncclient` [1]. This will allow automated deployment of measurement tests and remote management of their startup configurations.

4 Future Work

5 Conclusion

References

1. Bhushan, S., Tran, H., Schönwälder, J.: Ncclient: A python library for netconf client applications. In: Nunzi, G., Scoglio, C., Li, X. (eds.) IP Operations and Management, Lecture Notes in Computer Science, vol. 5843, pp. 143–154. Springer Berlin Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-04968-2_12`

[4] `http://code.google.com/p/libnetconf/`

2. Durand, A., Fasano, P., Guardini, I., Lento, D.: IPv6 Tunnel Broker. RFC 3053 (Informational) (Jan 2001), http://www.ietf.org/rfc/rfc3053.txt
3. Enns, R., Bjorklund, M., Schoenwaelder, J., Bierman, A.: Network Configuration Protocol (NETCONF). RFC 6241 (Proposed Standard) (Jun 2011), http://www.ietf.org/rfc/rfc6241.txt
4. Fainelli, F.: The OpenWrt embedded development framework. In: Proceedings of the Free and Open Source Software Developers European Meeting (2008)
5. Gilligan, R., Thomson, S., Bound, J., McCann, J., Stevens, W.: Basic Socket Interface Extensions for IPv6. RFC 3493 (Informational) (Feb 2003), http://www.ietf.org/rfc/rfc3493.txt
6. Huitema, C.: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380 (Proposed Standard) (Feb 2006), http://www.ietf.org/rfc/rfc4380.txt, updated by RFCs 5991, 6081
7. Thaler, D., Draves, R., Matsumoto, A., Chown, T.: Default Address Selection for Internet Protocol Version 6 (IPv6). RFC 6724 (Proposed Standard) (Sep 2012), http://www.ietf.org/rfc/rfc6724.txt
8. Wing, D., Yourtchenko, A.: Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555 (Proposed Standard) (Apr 2012), http://www.ietf.org/rfc/rfc6555.txt