Measuring TCP Connection Establishment Times of Dual-Stacked Web Services

Vaibhav Bajpai and Jürgen Schönwälder
Computer Science, Jacobs University Bremen, Germany
{v.bajpai, j.schoenwaelder}@jacobs-university.de

***Abstract*—The Internet Engineering Task Force (IETF) has developed protocols that promote a healthy IPv4 and IPv6 co-existence. The happy eyeballs algorithm for instance, provides recommendations to application developers to help prevent bad user experience in situations where IPv6 connectivity is broken. In this paper, we compare the IPv4 and IPv6 connectivity of a dual-stacked host using a metric that measures the Transmission Control Protocol (TCP) connection establishment time to a number of popular web services. We witnessed several cases where the connection establishment times and their variations over IPv6 are higher.**

I. Introduction

Research and corporate networks have had the capability to carry IPv6 traffic for a long time. However, due to the lack of IPv6 enabled content, the available infrastructure has rarely been used to access services outside of the internal network. With the World IPv6 day in 2011[1], this is starting to change with several notable web service providers enabling dual-stack mode to provide content over both IPv4 and IPv6. This has pushed network operators to develop deployment plans to bring IPv6 to residential customers. However, many network operators are still in a very early stage of deployment. As a consequence, early adopters that do not yet receive native IPv6 connectivity rely on tunnels to reach content over IPv6. Even the residential customers that do receive native IPv6 connectivity may experience performance and reliability issues, because the IPv6 deployed infrastructure may not be as mature as that of IPv4.

The IETF has identified these roadblocks and developed solutions to promote a healthy IPv4 and IPv6 co-existence. The happy eyeballs algorithm [1] for instance, provides recommendations to application developers to help prevent bad user experience in situations where IPv6 connectivity is broken. The algorithm, however, when combined with the default address selection policy [2], tends to give a noticeable advantage to connections made over IPv6. We want to know, given the state of the current IPv6 infrastructure, what is the amount of imposition a dual-stacked user has to pay by enabling IPv6 connectivity at home. In this pursuit, we have developed a metric to measure TCP connection establishment times. We use this metric to compare how TCP connection establishment times to a number of popular web services differ over IPv4 and IPv6.

[1]http://www.worldipv6day.org

The paper is organized as follows. In section II we survey studies on IPv6 adoption and its topology evolution. In section III we introduce our measurement methodology with a description of our measurement testbed in section IV. We capture our data analysis insights and conclude in section V.

II. Related Work

A number of studies have been conducted to measure the amount of IPv6 adoption on the Internet. Google has been collecting overall and country-based IPv6 adoption statistics[2] for a few years. The statistics reveal that, overall IPv6 adoption is increasing with a decrease in Teredo [3] and 6to4 clients. Hurricane Electric (HE) also maintains metrics[3] of global IPv6 deployment on the Internet, with statistics such as registered domains with AAAA records or networks with IPv6 support.

Sebastian Zander *et al.* in [4] propose a web-based technique using Google ads and custom Javascript snippets to measure IPv6 client capabilities. They witnessed that around 2% of the total connections used IPv6 in a dual-stacked environment, where a sample re-weighting technique reduced multiple biases to show a 20% increase in clients using happy eyeballs in their applications. The authors use this metric in [5] to further investigate Teredo [3] client capabilities. They show that around 16% of total client connections would be able to reach IPv6 services if Teredo capabilities in Windows were not reduced (Teredo in Windows cannot resolve service names to IPv6 endpoints). They also witnessed significantly higher latencies when using Teredo over native IPv4 or IPv6 connections. The metric based on Google ads is again used by Manish Karir *et al.* [6] in an extended seven-month long study to understand the amount and nature of IPv6 population on the Internet. They observed around 14M unique IPv6 addresses with native IPv6, Teredo, and 6to4 connectivity, and utilized the information embedded in IPv6 addresses to infer their geographical location, ISP, type of transition and NAT technology used. Hussein A. Alzoubi *et al.* in [7] study the performance implications of unilateral enabling of services over IPv6. They witnessed no performane penality in disabling the opt-in service. Google used to impose such an opt-in policy to allow hosts to receive Google services over IPv6. However, Google has recently changed the policy. This is discussed further in detail in section IV.

[2]http://www.google.com/ipv6/statistics.html
[3]http://bgp.he.net/ipv6-progress-report.cgi

ISBN 978-3-901882-53-1, 9th CNSM and Workshops ©2013 IFIP

MA	IPv4 Provider	IPv4 AS	IPv6 Provider	IPv6 AS	Location	Platform
1	German Research Network (dfn)	AS680	German Research Network (dfn)	AS680	Bremen	SamKnows (OpenWrt)
2	Kabel Deutschland	AS31334	HE (Tunnel-Broker)	AS6939	Bremen	SamKnows (OpenWrt)
3	Gaertner Datensysteme GmbH	AS24956	Gaertner Datensysteme GmbH	AS24956	Braunschweig	SamKnows (OpenWrt)
4	DTAG - Deutsche Telekom AG	AS3320	DTAG - Deutsche Telekom AG	AS3320	Bremen	SamKnows (OpenWrt)
5	BSKYB-BROADBAND-AS	AS5607	BSKYB-BROADBAND-AS	AS5607	London	SamKnows (OpenWrt)
6	ASN-IBSNAZ - Telecom Italia	AS3269	ASN-IBSNAZ - Telecom Italia	AS3269	Torino	SamKnows (OpenWrt)
7	BT ESPANA	AS8903	BT ESPANA	AS8903	Madrid	SamKnows (OpenWrt)
8	ROEDUNET	AS2614	ROEDUNET	AS2614	Timisoara	SamKnows (OpenWrt)
9	INIT7 - Init Seven AG	AS13030	INIT7 - Init Seven AG	AS13030	Olten	SamKnows (OpenWrt)
10	BT-UK-AS	AS2856	BT	AS5400	Ipswich	SamKnows (OpenWrt)
11	German Research Network (dfn)	AS680	German Research Network (dfn)	AS680	Bremen	Mac OS X
12	German Research Network (dfn)	A680	German Research Network (dfn)	AS680	Braunschweig	GNU/Linux
13	LambdaNet Communications	AS13237	Teredo	-	Berlin	GNU/Linux

TABLE I
LIST OF MEASURMENT AGENTS

III. METHODOLOGY

We have defined a metric that measures TCP connection establishment times. The metric also helps examine the impact of tunneling mechanisms employed by early adopters when reaching a dual-stacked web service. The metric essentially measures the time it takes to establish a TCP connection to a given endpoint. The input parameter is a tuple (service name, port number) and the output is the connection establishment time for all endpoints the service name resolves to, typically measured in microseconds.

The `happy`[4] program, a simple TCP happy eyeballs probing tool, is the implementation of our metric. The `happy` program starts by reading a list of service names either provided as command line arguments or from a file and uses `getaddrinfo(...)` to resolve the names to a list of IP endpoints. It then uses non-blocking `connect(...)` calls to concurrently establish connections to all endpoints of a service and measures the elapsed time along with an indication on whether the connection got established. The connection establishment indication is made once a socket in a `select(...)` call becomes writeable with no pending socket errors. It is important to note that the domain name resolution time is not accounted in the measured connection establishment time. The tool enforces a small delay (by default in the order of 25ms) between concurrent `connect(...)` calls to avoid bursty TCP `SYN` traffic and hence to improve accuracy. This enforced delay, however, does not obstruct the completion of any pending `connect(...)` calls. The tool also has the capability to lock the output stream to coordinate multiple writes to the same output stream. This is useful when multiple `happy` instances try to append results to a single regular file. We have cross-compiled `happy` for the OpenWrt platform, so that the tool can be deployed on SamKnows[5] probes.

By repeated execution of the `happy` program, we are able to collect time series of connection establishment times that provide us with insights on how IPv6 connectivity to services compares to IPv4 connectivity.

[4]http://happy.vaibhavbajpai.com
[5]http://www.samknows.com

IV. MEASUREMENT TRIALS AND DATA ANALYSIS

The `happy` tool must measure against a popular list of dual-stacked service names to capture the perspective of a dual-stacked host from a global standpoint. This will be required to capture the performance difference of the IPv4 and IPv6 infrastructure. We also need to measure from different locations of the Internet and we need to ensure that access to certain services is not blocked administratively.

A. Selection of Web Services

Alexa ranks and maintains listings of the most popular websites on the Internet. The public REST API, however, provides the capability to retrieve only the top 100 service names. Only a fraction of these top 100 service names are dual-stacked today. As such, the data made available via the REST API is just not enough to generate a top 100 dual-stacked service names list.

HE, a major IPv6 tunnel-broker based in the US, maintains a top 100 dual-stacked service names list[6]. They use the top 1M service names list made available by Amazon. However, we noticed that some of the popular web services (e.g. Wikipedia) are missing from this list even though they are dual-stacked. It appears some services provide `AAAA` records only for domain names starting with `www`. For example, `www.bing.com` does have a `AAAA` record while `bing.com` does not. (In this particular case, a request to fetch the latter leads to a redirect to the former) Since, HE does not follow `CNAME`s when processing the service names list, they miss some dual-stacked services in their top dual-stacked service list calculation.

We decided to use Amazon's top 1M service names list[7] used by HE as input to prepare a top 100 dual-stacked service names list using our own custom script. Our script prepends each service name with the label `www` to make an additional DNS request and it also explicitly follows `CNAME`s. This way, we do not miss any of the popular dual-stacked web services like `wikipedia.org`.

[6]http://bgp.he.net/ipv6-progress-report.cgi
[7]http://s3.amazonaws.com/alexa-static/top-1m.csv.zip

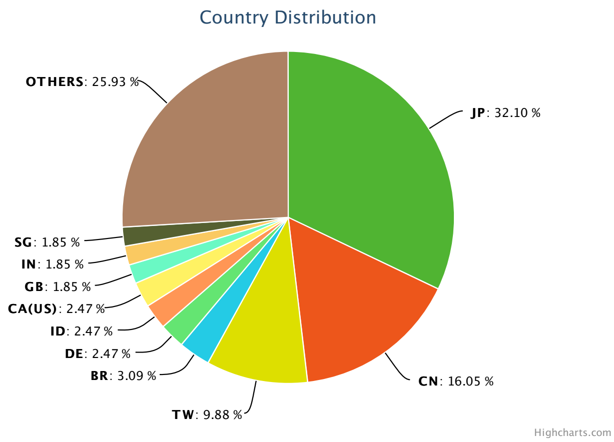

Fig. 1. A country-based distribution of prefixes blacklisted by Google over IPv6. The pie chart was drawn using the `highcharts.js` library.

B. Whitelisting and Blacklisting

Google used to perform `AAAA` prefix whitelisting to prevent users with broken IPv6 connectivity from requesting services over IPv6. Only the whitelisted DNS resolvers received `AAAA` records for Google services. This was an opt-in process, where an ISP explicitly signed up to receive Google services over IPv6. This helped ensure users had reliable IPv6 connectivity before trying to reach Google services over IPv6 [8].

Our top 100 dual-stacked web services list contains multiple services which either are owned by Google or are hosted on Google's infrastructure. Hence, it is necessary to ensure that the deployed Measurement Agent (MA)s would receive Google's services over IPv6, by capturing Google's list of whitelisted prefixes. Since the World IPv6 Launch Day in 2012[8], Google has changed their policy. The whitelist has been replaced by a blacklist[9]. This eliminates the opt-in process and increases the chance of a dual-stacked host reaching Google services over IPv6. However, if a host is behind a resolver from a blacklisted prefix, it will not receive Google services over IPv6 even though the host may enjoy perfect IPv6 connectivity from the network provider.

The pie chart in Fig 1 shows a country-based distribution of the blacklisted prefixes. The geolocation of the prefix is fetched from the GeoLite data created by MaxMind[10] and is derived from the announcements received from within the Border Gateway Protocol (BGP) routing system. The BGP routing data used is made publicly available by RIPE Network Coordination Centre (RIPE NCC)'s Remote Routing Collectors (RIS). It is possible that a prefix may be used from locations encompassing multiple countries. In such a case, the prefix is made to fall in the country with the highest coverage. Ideally, each location of the prefix should be accounted for to make the distribution more accurate. The google map in Fig 2 shows the location of the blacklisted prefixes from where they are announced into the BGP routing system. A large number of blacklisted prefixes appear to originate from Japan. These are ISPs whose DNS resolvers explicitly started filtering `AAAA` records after World IPv6 launch day and are now blacklisted.

Google's blacklist is dynamically changing. As a result, a backend scheduled job is provisioned to periodically update the raw data and regenerate the plots. The periodicity is currently set to a month. A webpage[11] has been created to keep the plots updated and allow further interactivity.

[8]`http://www.worldipv6launch.org`
[9]`http://www.google.com/ipv6/statistics/data/no_aaaa.txt`
[10]`http://www.maxmind.com`

C. Trial Deployments and Initial Insights

We ran `happy` on our internal test-bed of multiple MAs using our generated dual-stacked service names list as input. None of these MAs are behind blacklisted resolvers and therefore can receive Google services over IPv6. The MAs have different flavors of IPv4 and IPv6 connectivity ranging from native IPv4, native IPv6, IPv6 tunnel broker endpoints [9], Teredo [3] and tunnelled IPv4 as shown in Table I. The happy test was executed on the top 100 dual-stacked services list every 10 minutes and data was collected for a month.

The data collected by the measurements can be used to measure how the IPv6 connectivity of a dual-stacked host compares to the IPv4 connectivity when reaching a list of popular services. For instance, Fig. 3 shows the mean TCP connection establishment time and its standard deviation from one of the MA over 30 days. While several services show similar performance over IPv4 and IPv6, there are some services where there are notable differences and in general we observed that many MAs report higher variance over IPv6 compared to IPv4.

V. Conclusion

We have performed a preliminary study on measuring the TCP connection establishment times. The test was deployed on a number of MAs with varying flavors of IPv4 and IPv6 connectivity. We noticed several cases where the connection establishment times over IPv6 were higher. In order to develop a more comprehensive picture, it would be desirable to run this test on a large number of MAs attached to different IPv6 networks.

[11]`http://googleipv6.vaibhavbajpai.com`

Fig. 2. The geolocation of announced prefixes blacklisted by Google over IPv6. The map was drawn using the `gmaps.js` library.

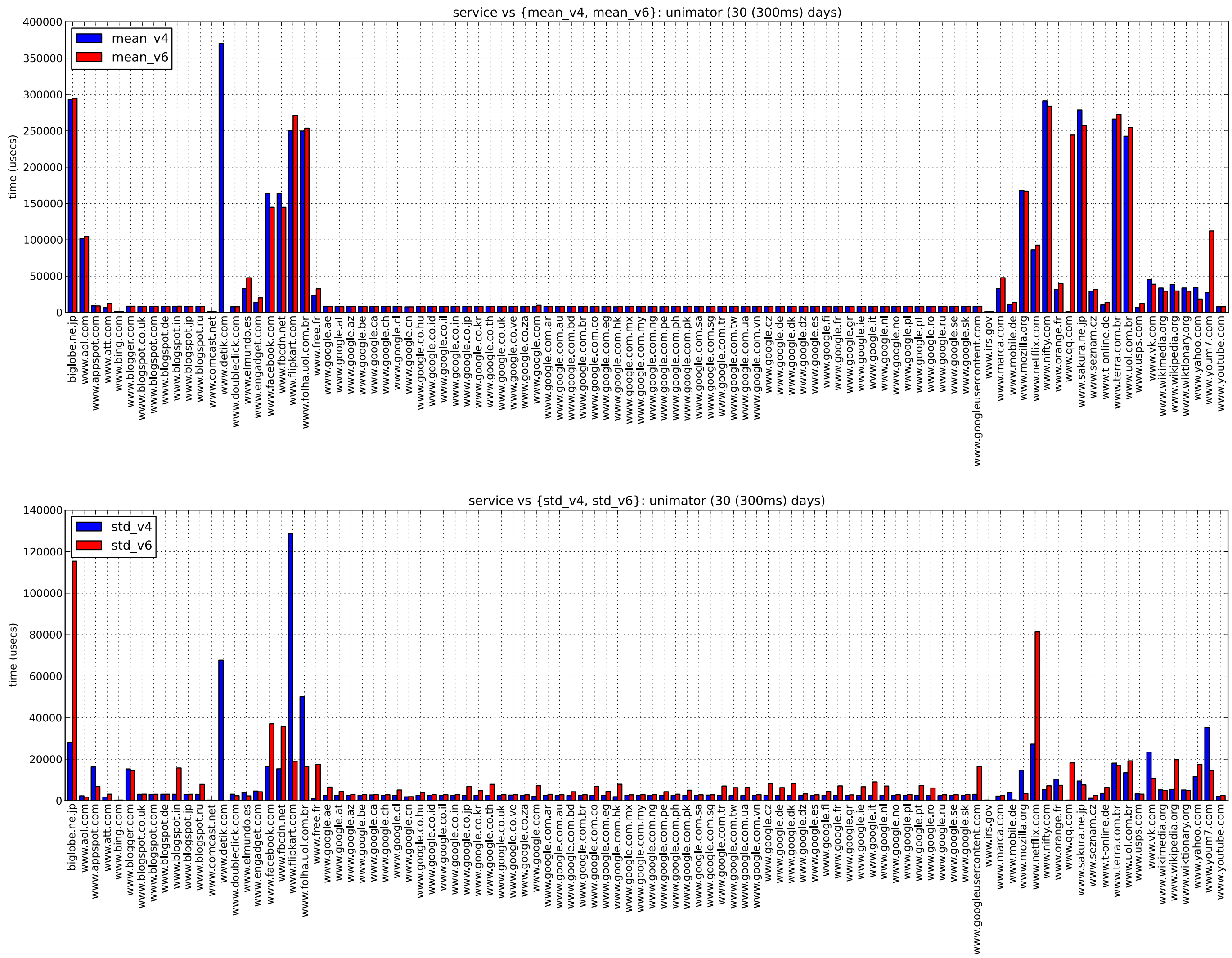

Fig. 3. Mean time and its standard deviations to establish TCP connections to a list of web services. The measurement agent is a server located at the University of Braunschweig. It has native IPv4 and IPv6 connectivity via the German Research Network [AS680].

VI. Acknowledgements

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

References

[1] D. Wing and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts," RFC 6555 (Proposed Standard), Internet Engineering Task Force, Apr. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6555.txt

[2] D. Thaler, R. Draves, A. Matsumoto, and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)," RFC 6724 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6724.txt

[3] C. Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," RFC 4380 (Proposed Standard), Internet Engineering Task Force, Feb. 2006, updated by RFCs 5991, 6081. [Online]. Available: http://www.ietf.org/rfc/rfc4380.txt

[4] S. Zander, L. L. Andrew, G. Armitage, G. Huston, and G. Michaelson, "Mitigating Sampling Error when Measuring Internet Client IPv6 Capabilities," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 87–100. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398787

[5] ——, "Investigating the IPv6 Teredo Tunnelling Capability and Performance of Internet Clients," *SIGCOMM Computer Communication Review*, vol. 42, no. 5, pp. 13–20, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2378956.2378959

[6] M. Karir, G. Huston, G. Michaelson, and M. Bailey, "Understanding IPv6 Populations in the Wild," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, M. Roughan and R. Chang, Eds. Springer Berlin Heidelberg, 2013, vol. 7799, pp. 256–259. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36516-4_27

[7] H. Alzoubi, M. Rabinovich, and O. Spatscheck, "Performance Implications of Unilateral Enabling of IPv6," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, M. Roughan and R. Chang, Eds. Springer Berlin Heidelberg, 2013, vol. 7799, pp. 115–124. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36516-4_12

[8] J. Livingood, "Considerations for Transitioning Content to IPv6," RFC 6589 (Informational), Internet Engineering Task Force, Apr. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6589.txt

[9] A. Durand, P. Fasano, I. Guardini, and D. Lento, "IPv6 Tunnel Broker," RFC 3053 (Informational), Internet Engineering Task Force, Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3053.txt