Mitigating Sampling Error when Measuring Internet Client IPv6 Capabilities

Sebastian Zander
CAIA, Swinburne University of Technology
Melbourne, Australia
szander@swin.edu.au

Lachlan L. H. Andrew
CAIA, Swinburne University of Technology
Melbourne, Australia
landrew@swin.edu.au

Grenville Armitage
CAIA, Swinburne University of Technology
Melbourne, Australia
garmitage@swin.edu.au

Geoff Huston
Asia Pacific Network Information Centre (APNIC)
Brisbane, Australia
gih@apnic.net

George Michaelson
Asia Pacific Network Information Centre (APNIC)
Brisbane, Australia
ggm@apnic.net

ABSTRACT

Despite the predicted exhaustion of unallocated IPv4 addresses between 2012 and 2014, it remains unclear how many current clients can use its successor, IPv6, to access the Internet. We propose a refinement of previous measurement studies that mitigates intrinsic measurement biases, and demonstrate a novel web-based technique using Google ads to perform IPv6 capability testing on a wider range of clients. After applying our sampling error reduction, we find that 6% of world-wide connections are from IPv6-capable clients, but only 1–2% of connections preferred IPv6 in dual-stack (dual-stack failure rates less than 1%). Except for an uptick around IPv6-day 2011 these proportions were relatively constant, while the percentage of connections with IPv6-capable DNS resolvers has increased to nearly 60%. The percentage of connections from clients with native IPv6 using happy eyeballs has risen to over 20%.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network Monitoring*; C.4 [**Performance of Systems**]: Measurement Techniques

Keywords

IPv6 deployment, banner-ad-based measurement

1. INTRODUCTION

In response to the foreseen exhaustion of IPv4 address space, the IETF standardised the IPv6 protocol as a long-term replacement for IPv4 in 1998 (for an introduction to IPv6 see [1, 2]). As of May 2012 most of the IPv4 address space is now allocated and according to predictions, the Regional Internet Registrars (RIRs) will run out of IPv4 addresses between 2012 and 2014 [3]. The question is: how many clients can access Internet resources with IPv6?

IMC'12, November 14–16, 2012, Boston, Massachusetts, USA.

Over the last decade several researchers studied the progress of IPv6 deployment [4]. Many of these studies were based on active probing of servers and network topology, or passive measurements based on server or traffic logs, routing data, or DNS root server data [5–11]. These provide valuable information about the IPv6 capabilities of servers or networks, and the proportion of clients that already use IPv6. However, these studies do not provide information about the IPv6 capabilities of clients. While most hosts run IPv6-capable operating systems, they are often not configured to use IPv6 (because there are virtually no IPv6-only services) or their access networks are not IPv6-capable.

More recently, a few researchers used active web-based measurements (based on a technique developed by S. Steffann [12]) to investigate the IPv6 capabilities of clients [13–15]. On certain participating web sites the clients' web browsers download not only the "normal" web pages and images, but also a few very small invisible images from a test server. One image can only be retrieved with IPv4, a second only with IPv6 and a third with either IPv4 or IPv6. Based on the images successfully retrieved one can determine the clients' IPv6 capabilities.

With this method the client sample has a bias based on the participating web sites. In previous studies only one or two web sites were used, or the analysis was limited to a region (e.g. EU countries). Also, the original method did not provide any means to verify whether a client actually attempted all image downloads or aborted some, for example because the user moved to a different web page quickly. Furthermore, previous studies focussed only on certain aspects (e.g. how many clients preferred IPv6 [13]).

We propose an improved version of the web-based measurement method, commonly implemented with JavaScript (JS-test). Our method not only measures the fraction of clients that prefer IPv6 over IPv4, but also the fraction of clients that are IPv6-capable (*including* latent Teredo [16] capabilities), or use an IPv6-capable DNS server for query resolution. We also analyse the capabilities based on the clients' operating system (OS) and country, and we estimate IPv6-related dual-stack failures and the use of "happy eyeballs" (fast fail-over from IPv6 to IPv4 [17]).

To increase the number of participating sites our test provides information not only to us, but also to the site operators themselves via Google Analytics. Since some web service providers are interested in the IPv6 capabilities of visiting clients, this has lead to an increase in participating sites. We also propose a novel method of selecting the clients to be tested. We embedded the test script in a

Flash ad banner (FA-test), which is served to clients via Google's AdSense. The test is carried out as soon as the ad is *displayed* by a browser. The ad's distribution is controlled by Google, but ads can potentially reach a very broad set of clients, through popular AdSense-enabled web sites (e.g. YouTube).

We compare the coverage of JS-test and FA-test and show that Google distributes the ads well. Normalised on the total number of tests, the FA-test reaches far more IP addresses located in more /24 networks compared to the JS-test. However, due the ad's running costs the number of FA-tests was limited, and the ad presentation was somewhat biased towards Asian web sites. Also, the FA-test could not test iPhones or iPads due to the absence of Flash.

We then identify sources of potential sampling error and propose techniques to mitigate it. We interpret our results as statistics of connections to avoid potential bias, for example due to multiple clients behind proxies or Network Address Translators (NATs). To mitigate the sampling error we re-weight the data based on the clients' countries and the proportion of Internet traffic of different countries. We compare the proportions of OSs and browsers measured with reference statistics and show that our raw statistics appear biased, but the re-weighted statistics are similar to the reference. We then estimate the IPv6 capabilities based on the re-weighted statistics.

The paper is organised as follows. Section 2 describes our measurement method. Section 3 describes the dataset and compares the client samples of JS-test and FA-test. Section 4 analyses sources of bias and presents our approach to mitigate the sampling error. Section 5 presents the IPv6 capability statistics measured. Section 6 discusses related work and compares it with our findings. Section 7 concludes and outlines future work.

2. MEASUREMENT METHOD

We describe the improved web-based measurement method, discuss the different ways tested clients are selected, and outline our experimental setup. Finally, we argue that our tests do not significantly affect the experience of users.

2.1 Web-based measurements

When users visit web sites their web browsers normally download several web pages, scripts, images etc. At certain participating web sites this includes a small test script that fetches a few invisible one-pixel images via HTTP from URLs pointing to our test web server (similar to a page hit counter). We refer to the script as *test* and a single image download as *sub-test*. For URLs containing host names the client's resolver has to first perform DNS look-ups against our authoritative test DNS server prior to sending HTTP requests. The following sub-tests allow us to test IPv4, IPv6, dual-stack, and (latent) Teredo [16] capabilities of clients, as well as whether the client's resolving DNS server is IPv6-capable:

1. The image can only be retrieved with IPv4 because the DNS server only returns an A record (IPv4-only).

2. The image can only be retrieved with IPv6 because the DNS server only returns an AAAA record (IPv6-only).

3. The image can be retrieved with IPv4 or IPv6 because the DNS server returns A and AAAA records (dual-stack).

4. The image URL is an IPv6 *address literal*. Windows Vista and Windows 7 hosts without native IPv6 will not query for DNS AAAA records, but with an IPv6 address literal they can use Teredo [18].

5. The image URL puts our authoritative DNS server behind an IPv6-only name server record. The DNS server can only be accessed if the client's resolving DNS server can perform DNS queries over IPv6.

A sub-test is deemed successful if the image could be retrieved, otherwise it is deemed a failure. After all sub-tests have been successfully completed or a timeout of ten seconds (whichever occurs first), the test script sends another HTTP request to the test web server that acts as a *test summary*. The test summary allows us to determine whether a browser has waited a sufficient amount of time for all sub-tests to complete. Without the summary it is impossible to know whether a sub-test was not successful because a client lacked the capability or because the test was interrupted.

The test script starts sub-tests in quick succession, but to which degree the images are fetched in parallel depends on the web browser's connection management. We assume that browsers try to fetch objects as quickly as possible without unnecessary delay. The URLs for each sub-test and summary resolve to different IPv4 and/or IPv6 addresses, which means browsers cannot multiplex different sub-tests over one TCP connection.

2.2 Client sample

Clients interact with our measurement system in two different ways. Several participating web sites link our JavaScript test script (JS-test) and visiting hosts are tested. To encourage participation in our JS-test, the script also logs the IPv6 capabilities of clients with Google Analytics and the site administrators can inspect the statistics (test script is available at [19]). The client sample is biased towards the participating web sites, but since the test is implemented in JavaScript the vast majority of visiting clients can be tested. We assume there are not many clients with disabled JavaScript [20].

We also implemented a Flash ActionScript test and embedded it in a Flash ad banner (FA-test). The ad is served to hosts via Google's AdSense. The test is carried out as soon as the ad is *displayed* by the web browser, the user does not have to click on it. We selected the ad's keywords and languages to achieve broad placement across different countries and sites. The FA-test reaches a more diverse client population, but cannot be carried out by clients without Flash, such as iPhones/iPads during our measurements.

2.3 Experimental setup

Figure 1 shows a logical diagram of our experimental setup. The DNS server handles the incoming DNS queries and the web server serves the test images. All servers are time-synchronised with NTP. Local Teredo and 6to4 relays are located close to the web server to reduce IPv6 tunnelling delay (Teredo server and client-side 6to4 relay are out of our control). We collect HTTP log data, DNS log data, and capture the traffic with tcpdump.

Several data fields are used to convey information from a tested client to our test servers, and to prevent caching of DNS records or test images at the client or intermediate caches (see Figure 1). The following data is prepended to the host name *as well as* appended as URL parameters: test time, test ID, test version, and sub-test name. Test time is the time a test started taken from the client's clock (Unix time plus milliseconds). Test ID is a "unique" random 32-bit integer number determined by the test script.[1] Test version is the test script's version number and sub-test name is a unique identifier for the sub-test, e.g. "IPv4-only".

[1]FA-tests use a hash value computed from Google's clickTAG, as Google prohibits the use of the ActionScript random() function.

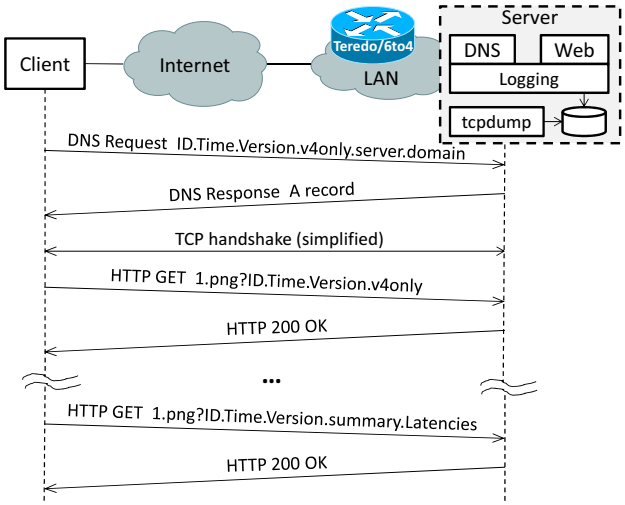

Figure 1: Experimental setup

The JS-test uses cookies to ensure clients are only tested once every 24 hours (per web site). The main reason for this is to reduce the load on our test servers. However, some clients may ignore the JS-test cookies and perform the test multiple times within 24 hours. The FA-test cannot use cookies due to Google's restrictions, so clients may be tested multiple times. For both tests there can be web proxies or NATs in front of multiple clients that look like repeating clients. Nevertheless, the number of IPs repeating the test within 24 hours is low (see Section 3.1)

2.4 User impact

The JS-test is only executed after the web page has loaded completely. Hence it does not delay the loading of the page. The JS-test script is 12 kB large (including the Google Analytics code). The loading of the FA-test is controlled by Google. Our ad is only 16 kB large, not larger than many other ads and well below Google's size limit of 50 kB. The test images are only 157 bytes large and invisible (loaded but not displayed). The total size of the test data transferred is well under 20 kB, less than 6% of the average web page size of 320 kB (as determined in [21]).

Our test opens six different TCP connections to download the test images (five for the sub-tests plus one for the summary). The download is done in the background by the browsers. The maximum number of concurrent open connections is at least 30 for most browsers [22]. Whether the test connections will cause existing connections to be closed depends on how many connections are already open. But in any case the test does *not* affect the loading of the current page and the next page will not load slower than when loaded for the first time.

We argue that overall our test does not have a significant impact on the user's browsing experience. We conducted a few simple tests in which test users did not experience any noticeable impact. Also, our test does not trick users into revealing any sensitive information. Instead, we utilise information that web clients normally send to *every* visited web server.

3. DATASET

We use the data collected between 16th of May 2011 and 19th of February 2012. We discarded 1% of tests as "invalid", such as tests with malformed or truncated URLs in the server logs. We also discarded tests of clients referred by three large web sites that only participated for one or two days. We analyse the data in blocks of days (24 hours). However, we will present most of the statistics averaged over weekly periods to reduce the error of the estimates. We now describe the properties of our dataset.

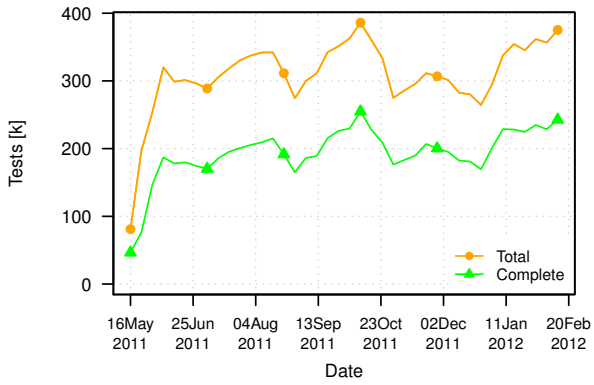

Figure 2: Total number of tests and number of completed tests per day

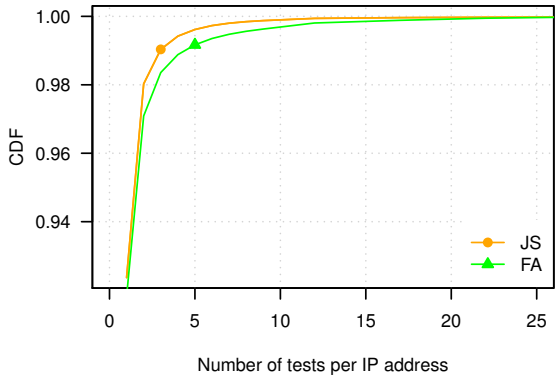

Figure 3: Number of tests per day per IP address for JS-test and FA-test

3.1 Number of tests

Figure 2 shows the weekly averages of the number of total and completed (with test summary) tests per day. Apart from the initial period the total number of tests per day was 250 000–300 000 and the number of completed tests was 180 000–200 000 per day (of which 30 000–35 000 were FA-tests and the rest were JS-tests).

The statistics we present in Section 5 are proportions. If the samples are independent, then the standard error for proportion estimates is $\mathrm{SE} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$, where $\hat{p}$ is the estimated proportion and n is the number of samples. Generally, for our statistics the relative error is below 1% and hence we do not plot error bars. However, due to the way we sample the clients there is an additional sampling error, which we discuss in Section 4.

Figure 3 shows the CDFs of the number of tests per day for each IPv4 address. FA-tests have a higher repeat rate (FA-tests cannot use cookies), but it is only slightly higher since Google distributed the ads well. Less than 8% of IPs performed the test more than once per day and less than 0.1% (JS-test) or 0.3% (FA-test) of IPs performed the test more than 10 times per day. This suggests that single IPs do not dominate our statistics.

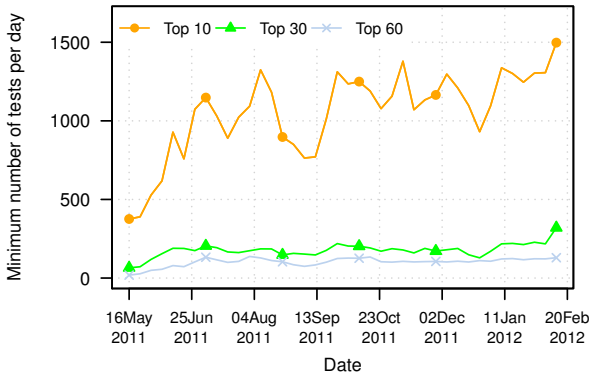

Figure 4: Minimum number of tests per day for the top-10, top-30 and top-60 traffic generating countries [24, 25]

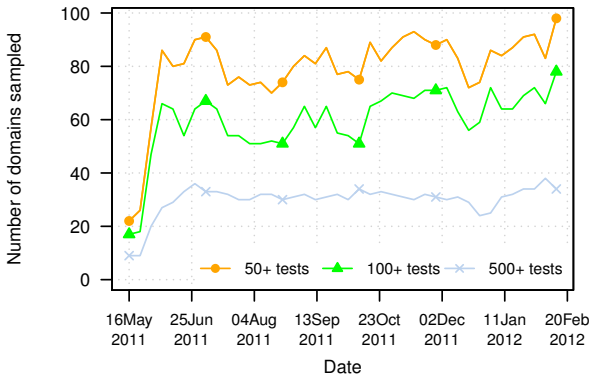

Figure 5: Number of domains with at least 50, 100 and 500 tests per day

3.2 Country and domain coverage

According to MaxMind's GeoIP [23] our dataset includes clients from 95% of all country codes including all 196 sovereign countries (the missing 5% are dependent territories). For about one third of the sovereign countries we observed at least 100 tests per day (with a few outliers), and these countries accounted for almost 100% of the Internet's traffic according to [24, 25]. Figure 4 shows the *minimum* number of samples per day per country for the top-10, top-30 and top-60 countries, which generate 76%, 90%, 98% of the Internet's traffic according to Cisco and Wikipedia statistics [24, 25] (we use the 5% quantile as minimum to exclude a few outliers). Apart from the initial period the minimum number of samples per day was higher than 750 for any of the top-10 countries, higher than 200 for any of the top-30 countries and at least 100–150 for any of the top-60 countries.

Figure 5 shows the number of domains sampled that generated at least 50, 100 or 500 tests per day (based on HTTP referrer information for JS-tests and Google ad placement information for FA-tests). Apart from the initial period, there always were 30–35 domains that each generated at least 500 tests a day and 55–75 domains that each generated at least 100 tests per day.

3.3 Client and subnet coverage

Figure 6 shows the number of unique versus total tested IPv4 addresses (left) and unique versus total tested /24 networks (right) for the measurement period (a /24 is obtained by setting an IPv4 address' last octet to zero). The dashed lines show the maxima (unique equals total). Since the IPv6 capability of clients depends on the clients themselves *and* on the clients' ISPs, a good spread over different /24 is also desirable.

The FA-test reaches a much larger proportion of unique IP addresses. The chance of the same IP address being tested multiple times is much lower than for the JS-test. It appears that Google tries to distribute the ads to a wide audience (at least in our case where the ad is not clicked often). For /24 networks the FA-test initially also tested unique /24 networks at a higher rate, but over time has slowed to a rate only slightly higher than that of the JS-test. The FA/JS-test sampled 0.40%/0.48% of the routed IP addresses and 22.2%/21.7% of the routed /24 networks (the total number of routed addresses divided by 256). Overall both tests sampled 0.75% of the routed IP addresses and 28.7% of the routed /24 networks. The percentages are based on Routeviews data [26] (2.52 billion routed addresses in February 2012).

Figure 7 plots the percentages of /24 networks from each /8 prefix where at least one IP was sampled for JS-test and FA-test versus each other. We classified the /8 prefixes based on which RIR mainly assigned them, and use the category "Several" for prefixes assigned by multiple RIRs. The figure shows that the JS-test covers significantly more /24s in many prefixes assigned by ARIN, and slightly more /24s in some prefixes assigned by RIPE or several RIRs (the latter mainly in the old class B space). On the other hand, the FA-test covers significantly more /24s in prefixes assigned by APNIC and a few more /24s in some prefixes assigned by LACNIC.

4. SAMPLING ERROR MITIGATION

Now we discuss sources of potential bias and present methods to mitigate the error. Our main method is based on re-weighting the data. We compare our OS and browser statistics from the raw and re-weighted data with reference statistics and show that the re-weighted statistics are a much better match than the raw statistics.

4.1 Error analysis and data re-weighting

Our data collection can be modelled as probability sampling, since random clients are tested. However, our client sample is not uniform. Different sites hosting the JS-test have different audiences. Hence, the JS-test error has two parts: a sampling error due to the random choice of sites and the bias of results conditional on a given site. We view the total error as sampling error since, averaged over all possible random choices of sites, the average per-site bias would be zero. The FA-test has an actual bias towards low-revenue sites showing our Google ad, and we abuse terminology by referring to this also as sampling error.

Bias may be caused by clients that do not respond to the test. The JS-test uses JavaScript, but all major browsers are JavaScript-capable and JavaScript is usually enabled as many web pages rely on it. According to [20] only 1–2% of clients have JavaScript disabled. Also, disabled JavaScript may be uncorrelated with IPv6 capability. We do not expect significant error to be introduced by this non-response.

The FA-test requires Flash, which may not be present on all clients. Most notably during the time of our experiments iOS (iPhone, iPad) did not support Flash, and on some Unix OS Flash also does not always work out-of-the-box. Since IPv6 capability depends on the OS, it is likely that the lack of Flash in non-Windows OS introduces bias. Furthermore, bias could be introduced by clients that use ad-blockers, which prevent the execution of ads. However, it is unclear what percentage of clients use ad-blockers and whether their use is correlated with IPv6 capability.

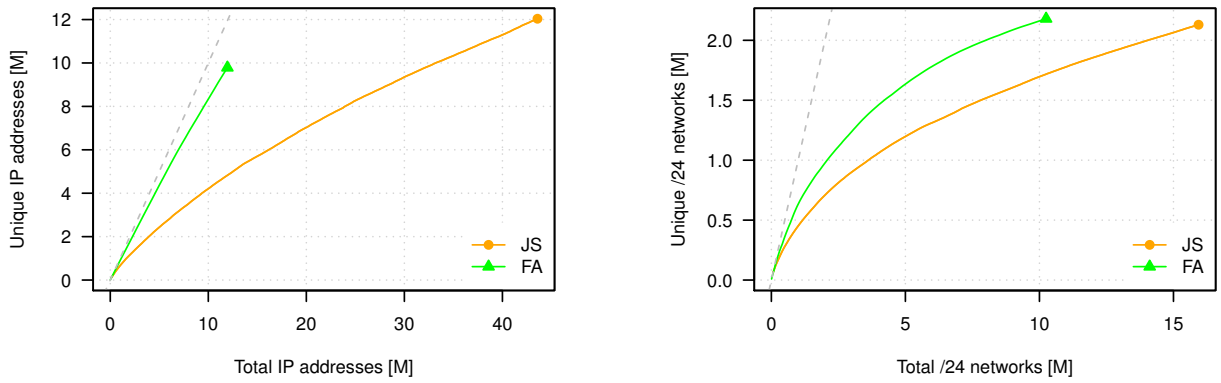

Figure 6: Unique vs. total tested IPv4 addresses (left) and unique vs. total tested /24 networks (right). The dashed lines show the maxima (all observed IPs or /24s are unique).

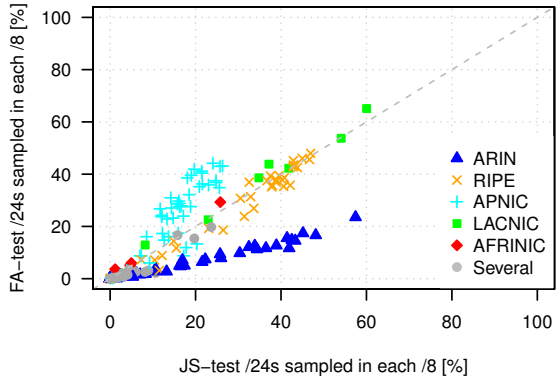

Figure 7: Percentage of /24 networks from each /8 prefix where at least one IP was sampled for JS-test (*x*-axis) and FA-test (*y*-axis)

Another source of potential bias are clients that do not perform some sub-tests, not because they lack the capabilities, but because of other reasons. For example, if a user moves to another web page quickly, this effectively aborts a test. To eliminate this bias we analyse only *completed* tests (with test summary). For these we can be sure that the clients attempted all sub-tests.

Our sample of tested clients is biased towards the web sites activating the JS-tests and FA-tests. For example, Indonesian JS-test connections are well over-represented due to a large participating Indonesian web site. Furthermore, the participating web sites change over time. To mitigate this bias we weight each test based on the tested client's country.[2] Let P_c be the weight of country c ($\sum P_i = 1$), T be the total number of tests, and T_c be the number of tests of country c . Then the weight for a test W_t is:

$$W_t = P_c \frac{T}{T_c} \ . \tag{1}$$

Our weights P_c are based on traffic statistics estimated by Cisco [24] for 16 countries that generate 79% of the Internet's traffic and Wikipedia country traffic statistics [25]. The traffic proportion ranks of the 16 countries in both datasets are broadly consistent. Spearman's rank correlation is 0.54 (1.0 indicates perfect positive correlation) and a hypothesis test indicates correlation (at 99% significance level). However, the Wikipedia data appears biased towards non-Asian countries. Hence, we use the Cisco estimates for the 16 countries covered and the Wikipedia statistics for the remaining countries.

We use MaxMind's GeoLite country database [23] to map IPv4 client addresses to countries, which has a claimed accuracy of 99.5%. Note that the ISO 3166-1 country codes represent not only sovereign countries, but also dependent territories and special areas of geographical interest.

Similar re-weighting approaches were used to mitigate bias before. For example, to deal with non-response the US Current Population Survey divided all sampled households into 254 adjustment cells and for each cell the sampling weight of non-respondents was redistributed to the other households in the cell [27].

We interpret the measurement results as *statistics of connections*[3] and not of clients. Interpreting the measurements as statistics of clients would result in an "observation bias". Also, IP addresses are not very good identifiers for clients (even when combined with browser ID strings). Using statistics of connections we avoid potential bias caused by multiple clients behind web proxies or NATs, or clients that change their IP addresses more often (home users, mobile users, frequently offline clients). Furthermore, we avoid potential bias because we are more likely to observe clients that are more likely to use the Internet, which potentially have more up-to-date systems and are more up to date with IPv6.

4.2 Effectiveness of re-weighting

To determine the possible bias in the raw data and the effectiveness of our re-weighting we compare the clients' OS, web browser, and Windows version distributions from our data with reference "population statistics". Since nobody has true population statistics we use the data of multiple statistics providers as reference.

We used the two-sample t-test (statistical hypothesis test) to compare the JS-test and FA-test combined raw and re-weighted data with the reference. The test determines whether the difference of the means of two samples is significant or due to random chance. The null hypothesis is that the difference between the two means is zero. If the null hypothesis is rejected, then there is a statistically significant difference.

[2]For <1% of JS-tests, where the IPs were proxies located in different countries than the clients (e.g. BlackBerry connections), we assume a client's country was the referring web sites' country (our data shows that many sites had relatively high locality).

[3]Here "connections" refers to instances of end-host connectivity and not to transport-layer connections.

Table 1: OS percentage of connections for raw (R) and re-weighted (W) data (mean±standard deviation)

	Reference [%]	JS R [%]	JS W [%]	FA R [%]	FA W [%]	Total R [%]	Total W [%]
Windows	81.5±4.5	89.1±1.8	83.5±1.6	96.6±1.0	93.1±0.8	90.3±1.8	86.2±0.9
MacOS X	8.8±3.1	3.3±0.5	8.0±0.4	2.9±0.9	6.1±0.8	3.2±0.4	7.5±0.3
iOS	4.4±1.4	1.7±0.2	3.3±0.5	0.0	0.0	1.4±0.2	2.5±0.4
Linux	1.2±0.4	2.1±0.5	4.3±1.3	0.5±0.1	0.8±0.2	1.8±0.3	3.1±0.6

Table 2: Windows 7, XP, Vista percentage of all Windows connections for raw (R) and re-weighted (W) data (mean±std. deviation)

	Reference [%]	JS R [%]	JS W [%]	FA R [%]	FA W [%]	Total R [%]	Total W [%]
Windows 7	44.3±4.3	41.5±2.9	48.6±3.5	38.0±6.8	42.7±4.2	40.9±3.3	45.6±3.1
Windows XP	40.3±5.2	51.8±1.5	41.3±2.6	52.7±7.7	42.8±3.2	51.9±2.2	42.8±2.4
Windows Vista	13.2±3.0	6.4±1.8	9.3±1.1	9.2±2.0	14.4±1.2	6.9±1.4	11.0±0.8

Table 3: Browser percentage of connections for raw (R) and re-weighted (W) data (mean±standard deviation)

	Reference	JS R [%]	JS W [%]	FA R [%]	FA W [%]	Total R [%]	Total W [%]
MS IE	39.9±8.4	18.4±4.1	35.8±2.3	49.9±10.3	52.4±2.2	23.5±4.6	42.3±1.6
Firefox	24.5±2.3	47.7±1.5	27.5±1.9	20.8±3.9	20.8±1.6	43.2±2.9	24.7±2.3
Chrome	21.9±4.9	22.9±2.5	22.6±2.3	24.4±5.8	21.0±2.1	23.3±3.1	21.3±1.8
Safari	7.0±1.6	6.0±1.3	9.6±1.4	2.2±0.5	4.2±0.5	5.3±1.1	8.1±0.7
Opera	2.3±1.0	3.8±1.1	2.9±0.2	2.7±1.0	1.4±0.2	3.7±1.0	2.4±0.2

Table 1 compares the percentages of OSs for JS-test, FA-test, and both tests combined for both the raw and re-weighted data with the reference data (unweighted average of AT Internet, Clicky, Net Market Share, StatCounter, StatOwl, W3Counter, WebmasterPro, Wikimedia from [28]). The FA test is biased towards Windows, as iOS could not run Flash and possibly Flash on Linux also did not work for some distributions/browsers. For both tests the MacOS X percentage is surprisingly low compared to the reference, but it may be that our observed web sites are less frequently used by MacOS X clients. The re-weighted statistics are much closer to the reference data, except that the JS-test appears to have a bias towards Linux (possibly due to the participating sites attracting a higher than average Linux user fraction).

According to the t-tests, for the combined raw data the null hypothesis must be rejected for all OSs (at 99% confidence level). In contrast for the re-weighted data we cannot reject the null hypothesis for Windows, MacOS X and iOS, but must reject it for Linux (at 99% confidence level). The re-weighted means are similar, except for Linux, which is over-represented.

Table 2 shows the percentages of Windows XP, Vista and 7 as percentages of all Windows connections for our data and the reference (as above from [28]). The raw data is biased towards more Windows XP and fewer Windows Vista and Windows 7 connections (e.g. for the JS-test a large Indonesian blog site had a high Windows XP fraction). But the re-weighted data is close to the reference data.

According to the t-test for the combined raw data the null hypothesis is rejected for Windows XP and Windows Vista, but not for Windows 7 (at 99% significance level). However, for the combined re-weighted data the null hypothesis cannot be rejected for any of the three Windows versions.

Table 3 shows the browser percentages of all connections and the reference data (based on unweighted averages of June 2011, December 2011 and March 2012 data from StatCounter, Net Applications, W3Counter, Wikimedia, Clicky from [29]). The raw JS-tests have a much higher percentage of Firefox and a much lower percentage of MS IE than the reference data. But the re-weighted numbers are much closer to the reference. As said above, the FA-test is biased towards Windows hence MS IE has a larger share and the other browsers have a smaller share (in particular Safari, which is not popular on Windows). Most of the re-weighted data is closer to the reference data, except the percentage of MS IE.

According to the t-tests for the combined raw data the null hypothesis is rejected for all browsers except Chrome (at 99% significance level). In contrast for the re-weighted data the null hypothesis cannot be rejected for any browser (at 99% significance level).

In conclusion, most of the OS and browser statistics from the raw data appear to have bias, while all statistics based on the re-weighted data are relatively similar to the reference, with the exception of Linux being over-represented. The results indicate that the re-weighting is very effective. Nevertheless, we cannot determine whether it is similarly effective for the IPv6 capability statistics.

4.3 Discussion and limitations

The per-country re-weighting mitigates the bias introduced because the proportions of tests observed from different countries differ significantly from the countries' traffic proportions. However, there is still a bias towards the clients visiting the particular set of web sites observed. Nevertheless, our set of web sites is relatively large and diverse, consisting of 55–75 different domains (universities, ISPs, gaming sites, blog sites) that refer at least 100 tested clients per day. Furthermore, we will show that a number of the statistics (but not all) from re-weighted JS-tests and FA-tests match well despite the totally different set of web sites.

Using per connection statistics instead of per client/IP statistics means we avoid additional observation bias. We showed in Figure 3 that in general our statistics are not dominated by some IPs that generated a huge number of connections. However, we encountered two daily statistics with overall low numbers (e.g. clients that preferred IPv6) that were largely dominated by a single IP address. We corrected these outliers manually (by ignoring the IPs).

5. IPV6 CAPABILITY ANALYSIS

First we analyse the overall and OS IPv6 capabilities. Then we compare statistics for JS-test and FA-test. Finally, we analyse per-

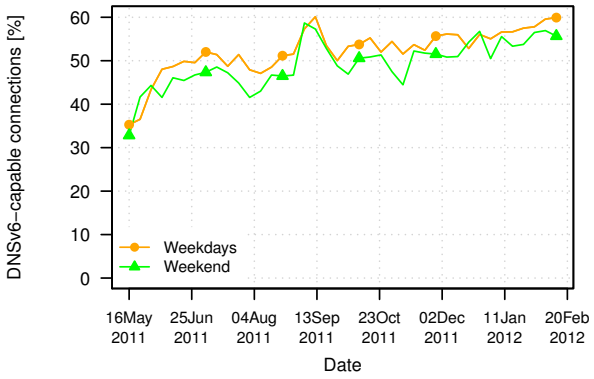

Figure 8: Percentage of all connections with IPv6-capable resolving DNS server

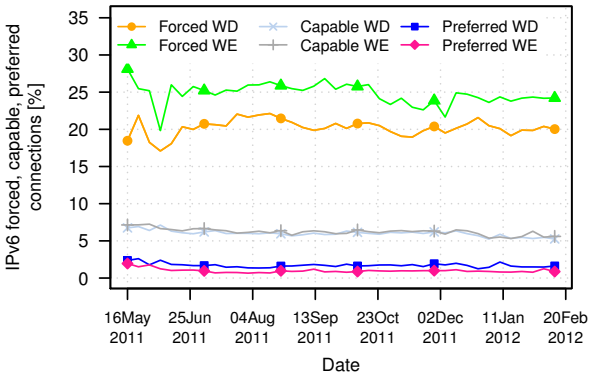

Figure 9: Percentage of all connections from hosts that were IPv6-capable, preferred IPv6 in dual-stack, or could be forced to use IPv6 with a literal URL

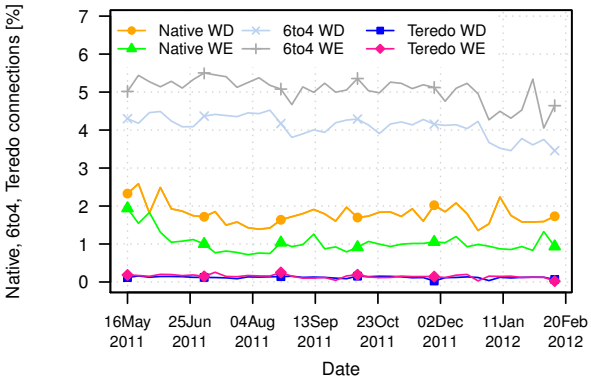

Figure 10: Percentage of all connections from IPv6-capable hosts by IPv6 type (native, 6to4, Teredo)

country IPv6 capabilities, dual stack failures and happy eyeballs usage.

5.1 Overall IPv6 capabilities

First we analyse the proportions of the connections where the client's resolving DNS server was IPv6-capable (DNSv6-capable), the client was IPv6-capable (IPv6-capable), the client preferred IPv6 over IPv4 in dual-stack (IPv6-preferred), or could be forced to use IPv6 with a literal URL (IPv6-forced). We plot the proportions for weekdays/workdays (WD) and weekends (WE).

Figure 8 shows the percentage of DNSv6-capable connections. The graph shows a clear uptrend from 40% to nearly 60% during our measurement period, meaning the percentage of connections from clients with resolving dual-stack DNS servers has increased. A slightly smaller weekend percentage suggests that fewer home user connections are DNSv6-capable.

Figure 9 shows the percentages of IPv6-capable, IPv6-preferred, and IPv6-forced. The percentage of IPv6-capable decreased slightly from 6–7% around World IPv6 day 2011 (Wednesday 8th of June), during which major Internet network and service providers enabled IPv6 as a test, to 6% with not much difference between weekdays and weekends. The percentage of IPv6-preferred connections decreased from its peak of 2.5% around IPv6 day to 1–2% in July 2011 and has remained relatively constant since then. The percentage of IPv6-preferred is higher during weekdays suggesting that home users were less likely to prefer IPv6 (probably because they were less likely to use native IPv6).

The percentage of IPv6-forced is basically the percentage of IPv6-capable plus 15–20% of connections originating from Windows Vista and Windows 7 hosts that support IPv6 via Teredo, but by default do not query for DNS AAAA records (see Section 2.1). The percentage of Teredo is almost 5% higher on weekends suggesting that Teredo is more likely used by home users. We think this is because Teredo is less likely to work in company networks due to being disabled or filtered by firewalls.

Only 6% of connections were from IPv6-capable clients, but for 11–12% of connections we observed DNS AAAA requests. We think the difference is mainly due to hosts that request AAAA records even if they have no IPv6 interfaces. For example, some older Firefox browsers cause this and it is a common default on Linux [30]. However, IPv6 failures may also contribute to the difference. A detailed analysis is future work.

Figure 10 shows a breakdown of the IPv6-capable connections into those using "native" IPv6 (including point-to-point tunnels and 6rd [31]), 6to4, or Teredo based on the IPv6 address prefixes.[4] Over 70% of the connections from IPv6-capable hosts used 6to4 tunnelling (4–5% of all connections), 20–30% of IPv6-capable connections came from hosts with native addresses (1–2% of all connections), and only 2–3% of IPv6-capable connections used Teredo – manually fully-enabled Windows Teredo or Teredo on other OS (0.1–0.2% of all connections).

Native IPv6 peaked in the week of IPv6-day, then decreased slightly and has remained relatively steady since (with a possibly very small uptrend). The percentage of 6to4 has decreased slightly over time. On weekends there are significantly fewer connections from hosts using native IPv6, but more connections from hosts using 6to4. During the week this reverses. This suggests a higher native IPv6 usage at work places.

Figure 11 shows that 90–100% of connections from hosts with native IPv6 prefer IPv6 in dual-stack. However, the percentage has decreased over time. We suspect this happened because of an increased use of happy eyeballs, which fails over very quickly from IPv6 to IPv4 (see Section 5.6). Only 0.5–1% of IPv6-capable 6to4 connections preferred IPv6 in dual-stack, and well under 0.1% of Teredo connections preferred IPv6 in dual-stack (Teredo omitted in Figure 11).

[4]The 6to4 protocol uses the prefix 2002::/16 and the Teredo protocol uses the prefix 2001:0::/32 [1, 2]. We ignore older IPv6-over-IPv4 tunnelling technologies since they are not common [5, 13].

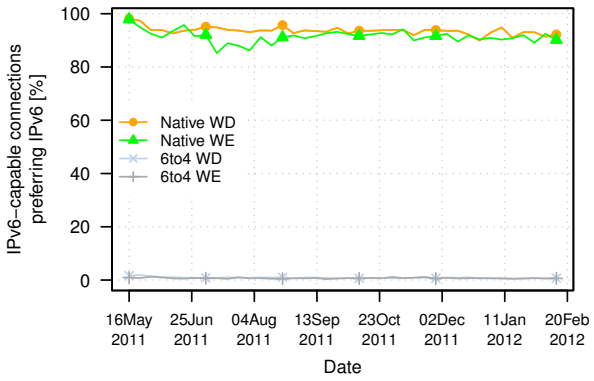

Figure 11: Percentage of IPv6-capable connections that preferred IPv6 in dual-stack (native IPv6, 6to4). Capable clients with Teredo never preferred IPv6.

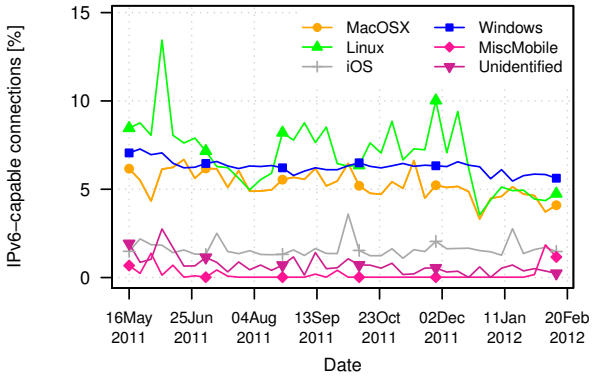

Figure 12: Percentage of all connections from IPv6-capable hosts by operating system

5.2 Operating system IPv6 capabilities

Figure 12 shows the percentages of IPv6-capable connections and Figure 13 shows the percentage of connections from IPv6-capable clients that preferred IPv6 for different OSs.[5] Over 6% of connections were from IPv6-capable Windows clients, but only for 20% of these (with native IPv6) the clients actually preferred to use IPv6 in dual-stack. In contrast MacOS X and Linux clients have a similar 6–7% of IPv6-capable connections, but most of them were native IPv6 and preferred to use IPv6 in dual-stack. Only 1–2% of connections came from IPv6-capable iOS clients, but about 30–40% (with native IPv6) preferred IPv6 in dual-stack.[6]

Figure 12 shows a significantly increased IPv6-capability for Linux around IPv6-day 2011. Many of the connections came from Ubuntu clients that enabled IPv6 via point-to-point tunnels on that day and even for a few weeks afterwards. In the raw data the spike is even higher (up to 30%), since multiple of our participating web sites were monitored by a company that uses Ubuntu-based probes located in different countries. We filtered out these probes as they are not "genuine" clients.

[5]We distinguish between MacOS X, Linux, BSD variants, Solaris, iOS, Windows, mobile OS (Symbian, BlackBerry, and Android), and unidentified OS. OSs with too few occurrences are not shown.

[6]Note that due to the low number of observed IPv6-capable iOS connections the standard error here is 5–10%.

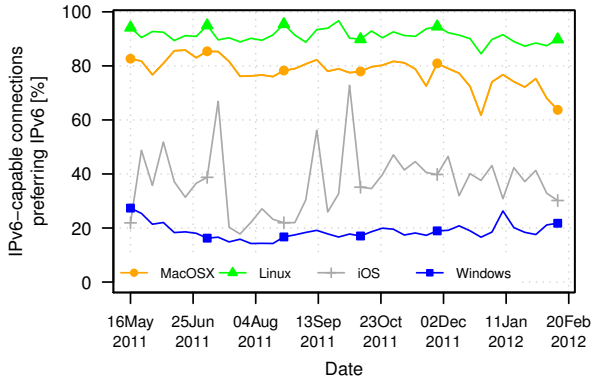

Figure 13: Percentage of IPv6-capable connections where the host preferred IPv6 by operating system

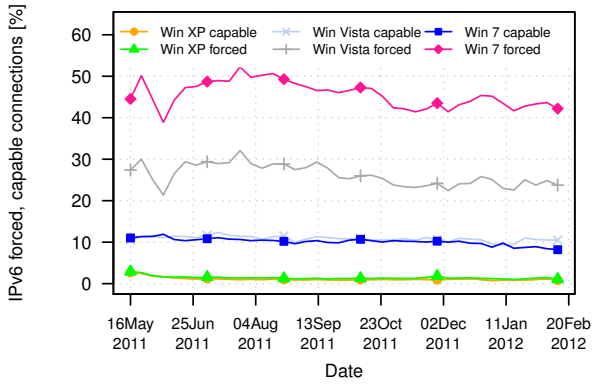

Figure 14: Percentage of IPv6-capable and IPv6-forced for Windows XP, Vista, and 7 connections

Figure 14 shows the percentages of IPv6-capable and IPv6-forced connections for all connections of each Windows version (Windows XP, Vista, and 7 only, as 99% of the tested Windows clients ran these versions). Only 1% of connections from Windows XP hosts were IPv6-capable and the percentage of IPv6-forced was equally low (6to4 and Teredo are not enabled by default on Windows XP). Windows Vista and Windows 7 had a similar 10% of IPv6-capable connections, but the percentage of IPv6-forced was 20% larger for Windows 7 (6to4 and Teredo are enabled by default on Windows Vista and 7, but the lower IPv6-forced percentage suggests that Teredo was either disabled or failed to establish a tunnel more often on Vista [32]).

The percentages of connections from IPv6-capable Windows XP, Vista and 7 clients that preferred IPv6 in dual stack were 20–30%, 10%, and 20% respectively (since there were no significant trends we omitted the figure). Windows XP has the highest IPv6-preferred (native IP) percentage, which is not very surprising since in contrast to Windows Vista or 7 6to4 is not enabled by default. The IPv6-preferred percentage is significantly higher for Windows 7 than for Windows Vista, maybe because more IPv6-savy Windows users avoid Vista.

5.3 Client sample dependencies

Figure 15 compares the IPv6-forced, IPv6-capable and IPv6-preferred proportions for JS-tests and FA-tests. IPv6-forced was higher for FA-tests since the proportion of Windows was higher.

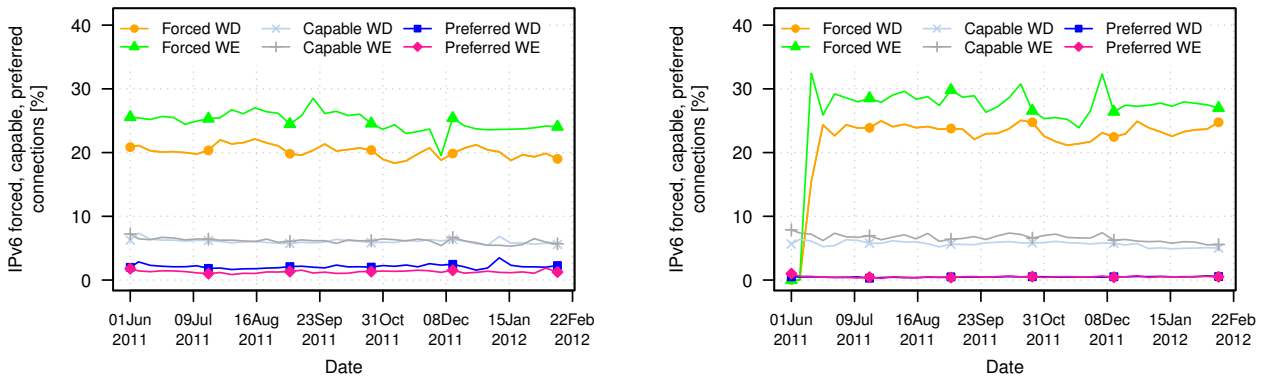

Figure 15: Percentage of IPv6-forced, IPv6-capable and IPv6-preferred for JS test (left) and FA test (right)

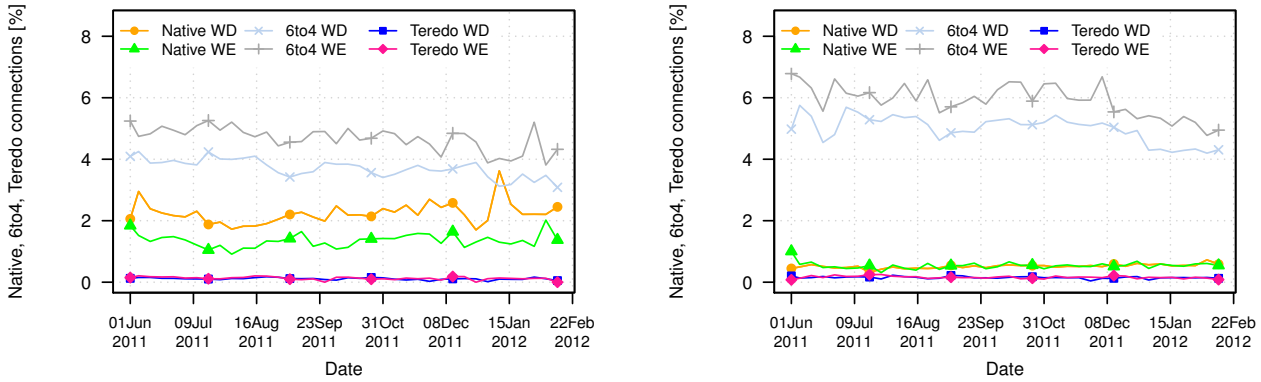

Figure 16: Percentage of all connection from IPv6-capable hosts by IPv6 type for JS test (left) and FA test (right)

IPv6-capable was similar for both tests (for FA-tests it was slightly lower and the difference between weekday and weekend was larger). IPv6-preferred was lower for FA-tests. FA-tests covered fewer MacOS X and Linux connections, the two OSs that strongly preferred IPv6, but also the IPv6-capable proportion for these two OSs was lower for FA-tests.

Figure 16 shows the breakdown of the IPv6-capable connections for JS-tests and FA-tests. The percentage of 6to4 was higher for FA-tests, whereas the percentage of native IPv6 was higher for JS-tests. However, for both tests the 6to4 percentage decreased slightly over time and the native IPv6 percentage peaked around IPv6 day, then decreased to a low in July 2011 and since then has increased very slightly. The weekday/weekend difference for 6to4 was consistent across both tests, but the weekday/weekend difference for native IPv6 was different. For JS-tests the percentage was significantly higher on weekdays, but for FA-tests there was no difference between weekdays and weekends. We suspect some JS-test web sites attracted IPv6-capable hosts during the week (e.g. APNIC web site), whereas the FA-test was more biased towards home users.

For both JS-tests and FA-tests 90–95% of connections from clients with native IPv6 addresses preferred IPv6 in dual-stack, but only 0.5–1% of 6to4 connections and under 0.1% of Teredo connections preferred IPv6 in dual-stack.

5.4 Country IPv6 capabilities

We now analyse the proportions of IPv6-capable and IPv6-preferred connections for the top-12 countries in each category over the whole measurement period. The number of Teredo connections does not vary much for different top-12 countries, hence we omit an analysis for brevity. We analysed JS-tests and FA-tests separately, but here focus on the results for FA-tests, since we think the JS-test is more biased because of the participating web sites.

Figure 17 shows the percentages of IPv6-capable connections for the FA-test for Brazil (BR), France (FR), Indonesia (IN), Italy (IT), Japan (JP), Korea (KR), Poland (PL), Romania (RO), Russia (RU), Taiwan (TW), Ukraine (UA), and the USA (US). For the JS-test the percentages and order of the countries differ, but 9 of the 12 countries (75%) in the top-12 are identical.

Figure 18 shows the percentage of IPv6-preferred connections for the FA-test for China (CN), France (FR), Germany (DE), Great Britain (GB), Japan (JP), Malaysia (MY), Poland (PL), Russia (RU), Taiwan (TW), Thailand (TH), Ukraine (UA), and the USA (US). Again, for the JS-test the percentages and order of the countries differ, but 8 of the 12 countries (66% of countries) in the top-12 are identical. Colitti *et al.* [13] analysed the world's top-10 countries of clients that preferred IPv6 previously. While their ranking differs from ours we note that 6 (JS-test) or 7 (FA-test) of their top-10 countries are also in our top-12.

5.5 Dual-stack failures

It would be interesting to know the proportion of connections that failed the dual-stack sub-test, because the clients tried to retrieve the test image unsuccessfully over IPv6. However, we cannot differentiate reliably between IPv6-related dual-stack failures

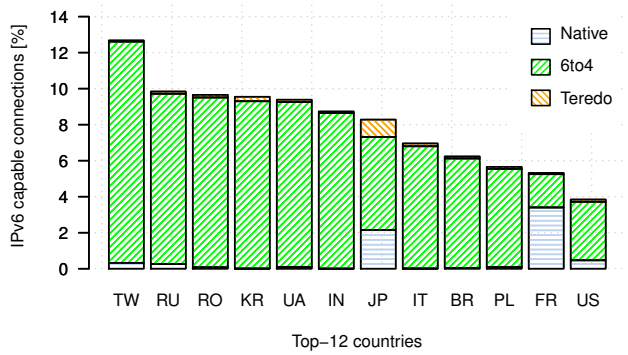

Figure 17: Percentage of IPv6-capable connections of top-12 countries for FA-test

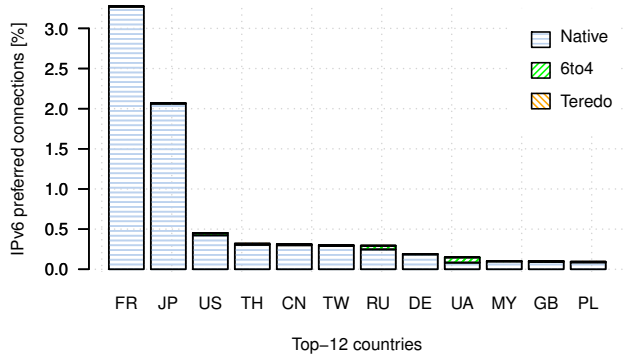

Figure 18: Percentage of IPv6-preferred connections of top-12 countries for FA test

or other connectivity failures. Nevertheless, we can estimate an upper bound.

Let e^{A}_{dual} be the dual-stack sub-test failure rate for connections where only a DNS A request was observed, e^{A+AAAA}_{dual} be the dual-stack sub-test failure rate for connections where both A and AAAA requests were observed, and $e^{A+AAAA}_{v4only+dual}$ be the proportion of connections with failed IPv4-only *and* dual-stack sub-tests for clients that asked for A+AAAA records. We assume that non-IPv6 related failures do not depend on whether clients asked for A+AAAA records or A record only. We estimate the proportion of dual-stack failures because of failed IPv6 to:

$$\varepsilon = e^{A+AAAA}_{dual} - e^{A}_{dual} - e^{A+AAAA}_{v4only+dual} \,. \qquad (2)$$

Still some of these failures may have other causes than failed IPv6, hence our estimated proportion is an upper bound. Figure 19 shows the estimated dual-stack error rate because of failed IPv6 (we only logged DNS requests since mid July 2011). Overall, well under 1% of connections with AAAA requests failed the dual-stack sub-test because of IPv6, but the error rate increased during our measurement.

5.6 Happy eyeballs

When a dual-stack client resolves a host name to IPv6 and IPv4 addresses, it will often try to reach the destination host using IPv6 first. If the client cannot reach the destination with IPv6, it will fail back to use IPv4. With older browsers/OSs this fail-over could take a long time (sometimes up to a minute). "Happy eyeballs" [17] enables browsers/OSs to reduce connection-setup times by minimising fail-over delays. They fail over to IPv4 very quickly if IPv6 connections cannot be established.

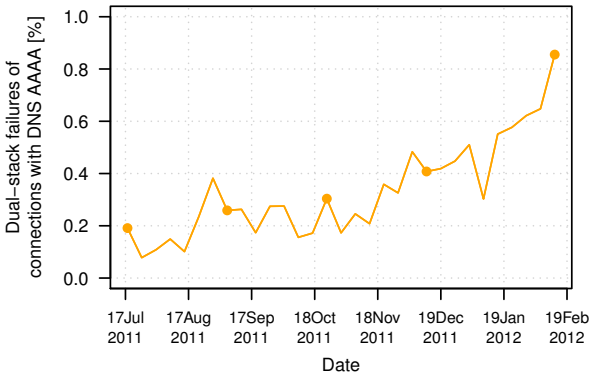

Figure 19: Percentage of connections with DNS AAAA requests with dual-stack failure due to IPv6 problems (upper bound)

We estimated the percentage of clients with native IPv6 that use happy eyeballs. If for a dual-stack sub-test a host tried to open IPv4 and IPv6 TCP connections simultaneously (within a one second time window) and one connection succeeded while the other was abandoned, we count this as happy eyeballs attempt. Figure 20 shows that we observed happy eyeballs for 10–20% of hosts with native IPv6 and the percentage increased. For 75% of these connections clients preferred IPv6 over IPv4, but for 25% of them clients failed over to IPv4.

Figure 21 shows the web browsers that used happy eyeballs (according to the browser ID strings). Most of the happy eyeballs came from hosts using Chrome, which has had working happy eyeballs support since May 2011 [33]. A smaller number of happy eyeballs came from Firefox and Safari. Since July 2011 MacOS X Lion had happy eyeballs support and the combination of Safari and MacOS X Lion works [34]. According to our data there were happy eyeballs from Safari running on MacOS X and on iOS.

Firefox had a happy eyeballs implementation since version 7, but prior to version 10 it was disabled by default [34]. Also, the initial Firefox implementation had problems; it was effectively removed in September 2011 and replaced by an improved version in November 2011 [35]. This explains the period without happy eyeballs from Firefox. Happy eyeballs was turned on by default in Firefox 10 (released end of January 2012), which is clearly visible in the graph.[7]

We also computed the average latency between the IPv6 and IPv4 connection requests. For Chrome it was on the order of 200–250 ms, close to the nominal 300 ms fail-over time [34]. For Firefox since version 10 the delay was also on the order of 200–250 ms, whereas for Safari the delay was 150–200 ms.

6. RELATED WORK

A number of researchers studied the progress of IPv6 deployment. The survey paper [4] provides an excellent overview and also identifies areas that need further study.

Some studies examined routing or DNS data, or used active measurements to identify the IPv6 capabilities of domains and

[7]Firefox automatically updates on a daily basis, assuming a user restarts it on request or reboots. Hence, an update can relatively quickly change the abilities of the Firefox population.

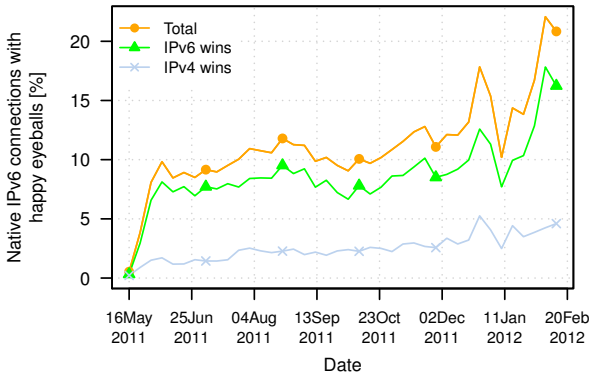

Figure 20: Percentage of connections from clients with native IPv6 that used happy eyeballs – overall and by winning protocol (IPv4 or IPv6)

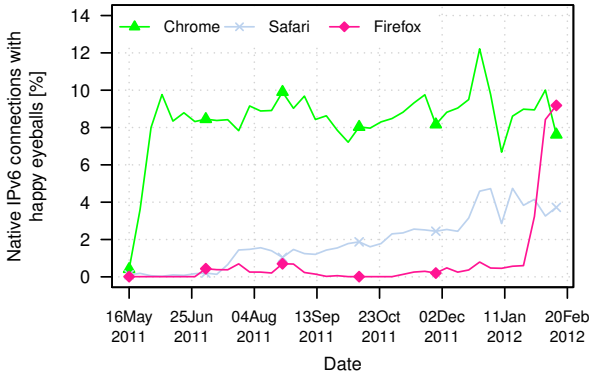

Figure 21: Percentage of connections from clients with native IPv6 that used happy eyeballs by web browser

servers. They do not provide any insights into the IPv6-capabilities of clients and we only briefly mention them. Leber [6] tracked the percentage of Autonomous Systems (ASs) that advertised IPv6 prefixes, and the percentage of top level domains and top-1000 web servers (based on Alexa) that were IPv6-capable (had DNS AAAA records). Prior [7] monitored the IPv6 capability of web, email, DNS, and NTP servers of certain domains. Kuehne [8] surveyed the IPv6 deployment by querying web and email servers of the top domains from Alexa for DNS AAAA records. Nikkah *et al.* [36] compared the performance of IPv4 and IPv6 based on accessing a larger number of IPv6-capable web servers. CAIDA [37] visualised the IPv6 Internet topology based on routing information and active probing.

Other studies analysed server logs and traffic traces. They only reveal the fraction of clients that preferred IPv6 over IPv4, but they cannot reveal the IPv6 capabilities of all clients. Also, some of these studies did not analyse the former, but focussed on other aspects, such as the IPv6 technologies used. Karpilovsky *et al.* [9] analysed the proportion of different IPv6 technologies used based on NetFlow data, but they did not analyse the overall proportion of IPv6 traffic. Shen *et al.* [10] studied the IPv6 deployment and the applications using IPv6 in China based on NetFlow records. They focused on the IPv6 techniques and addressing schemes used, but did not estimate the proportion of IPv6 traffic. Malone [5] analysed the proportion of IPv6 addresses from three server logs (Web, FTP, DNS) in 2007 and found that 0.3–0.4% of the connections used IPv6. Labovitz [11] measured an IPv6 traffic proportion of 0.03–0.04% in 2010 based on flow measurements from 100 Internet Service Providers (ISPs). Defeche and Wyncke [38] studied the use of IPv6 in the BitTorrent file sharing community and found that the fraction of IPv6-capable peers was 1%. The IPv6 proportions observed in [5, 11, 38] span almost two magnitudes, probably because each study observed a very specific set of clients.

Recently, a few studies used a web-based measurement method to study the IPv6 capabilities of clients based on Steffann's [12] approach. Colitti *et al.* [13] measured the proportion of hosts that preferred IPv6 and the IPv6 technologies used for different countries based on a fraction of Google users in 2008–2009. They found that only 0.25% of hosts used IPv6. Kreibich *et al.* [30] analysed Netalyzr logs and reported that 4.8% of sessions used IPv6. However, they note that this is an upper bound due to possible caching effects and "geek bias" (Netalyzr only tests hosts of users who choose to be tested). Smets *et al.* [14] estimated the IPv6 deployment for EU states based on over 30 participating web sites in 2010. They found that the proportion of IPv6-capable hosts varied from under 1% (Cyprus) to over 12% (Sweden) with an average of 6%. Aben [15] measured the IPv6 deployment based on DNS AAAA records for popular servers, advertised IPv6 routing prefixes, and active web-based measurements embedded in the RIPE web site. Aben found that in 2011/2012 around 6–7% of clients were IPv6 capable and 2–3% of clients preferred IPv6. Anderson [39] measured a client dual-stack failure rate of 0.015% at two web sites in Norway in May 2011.

We measured 6% of IPv6-capable connections, which is consistent with the 6% observed in [14] (for EU countries) and the 6–7% measured in [15]. We measured 1–2% of connections that preferred IPv6 over IPv4. This is much higher than the 0.25% measured in [13], but lower than the 2–3% reported in [15] and the 4.8% upper bound reported in [30]. Our dual-stack failure rate is much higher than the rate reported in [39], because our proportion is based on the connections with DNS AAAA requests and not on all connections. As proportion of all connections our estimate is 0.02–0.09%.

Compared to most previous studies we sampled a wider range of clients and domains, including clients sampled by our novel Google-ad based measurement approach. Also, we provide a more comprehensive analysis, e.g. we distinguish between IPv6-preferred and IPv6-capable (including latent Windows Teredo) clients, and analyse the IPv6 capabilities of the clients' resolving DNS servers. Furthermore, only our study analyses the client sample properties, discusses the sampling error and proposes methods to mitigate the error.

7. CONCLUSIONS AND FUTURE WORK

Despite the predicted exhaustion of unallocated IPv4 addresses in the next two years, it remains unclear how many clients can use its successor IPv6. We proposed a refined web-based measurement approach that mitigates intrinsic measurement biases, and demonstrated a novel web-based technique using Google ads to perform IPv6 capability testing on a wider range of clients. To mitigate the sampling error we re-weighted the raw data. We compared raw and re-weighted OS and browser statistics with reference statistics and showed that our raw statistics appear biased, but the re-weighted statistics are similar to the reference. We then used the re-weighted statistics to estimate the IPv6 capabilities of clients.

We observed slightly over 6% of connections from IPv6-capable clients, but over 70% of these were from Windows clients that used 6to4 tunnelling and did not prefer IPv6 in dual-stack. Only 1–2% of connections were from clients with native IPv6, but over 90% of

them preferred IPv6 in dual-stack. The dual-stack failure rate (as percentage of connections with DNS AAAA requests) was under 1%, but we observed an increasing trend. The use of happy eyeballs (fast fail-over from IPv6 to IPv4) rose to over 20% of connections from clients with native IPv6.

The percentage of connections with IPv6-capable resolving DNS servers increased significantly from 40% to nearly 60%. On the other hand, the percentage of connections from IPv6-capable clients and the percentage of clients that preferred IPv6 peaked around IPv6 day 2011, decreased slightly and then remained relatively constant. The IPv6-capable percentage actually decreased slightly. However, this decrease was caused by a decline in 6to4 connections, while the percentage of native IPv6 connections increased very slightly. Overall, the trends suggest that the adoption of IPv6 by clients is still very slow, at least in the client sample we observed. However, the increasing percentage of connections with IPv6-capable DNS servers is encouraging, as it could indicate an increased number of dual-stack DNS servers and possibly an increased IPv6-capability of ISPs.

We noticed a clear difference between weekdays and weekends, most likely driven by the different capabilities of work and home clients. The proportion of native IPv6 connections was higher during the week, whereas the proportion of 6to4 and Teredo was higher on weekends. The proportion of IPv6-capable connections was roughly similar for Windows, MacOS X and Linux. However, while 80% of the Windows connections relied on 6to4, most MacOS X and Linux connections used native IPv6.

An additional 15–20% of all connections from Windows 7 and Windows Vista clients could use IPv6 with Teredo. However, by default the Windows resolver does not ask for DNS AAAA records if the only IPv6 network interface is a Teredo interface, so these clients are not truly IPv6-capable. Without Teredo only 10% of connections from Windows Vista or Windows 7 clients were IPv6-capable, but with Teredo the percentage increased to 20–30% for Windows Vista and 40–50% for Windows 7.

Our Google ad based measurement technique could be used for other types of client-based measurements, if they can be performed with the functionality of ActionScript available in Google Flash ad banners. Our current test script already measures the delays for fetching the test images [32], and this information could possibly be used to analyse clients' network access technologies. Another possibility is to estimate the bottleneck capacities of clients with a tailored packet-pair measurement approach. Our proposed reweighting method could be applied to other problems where a geographically biased sample of Internet clients is observed.

In the future we plan to increase the sample size, estimate the size of the sampling error, and improve the sampling error reduction. We also plan to extend our measurement methodology, so that we can determine IPv6 failures more accurately and identify their causes.

Acknowledgements

We thank our shepherd Kenjiro Cho and the anonymous reviewers for their helpful comments. This research was supported under Australian Research Council's Linkage Projects funding scheme (project LP110100240) in conjunction with APNIC Pty Ltd and by Australian Research Council grant FT0991594.

8. REFERENCES

[1] I. van Beijnum. *Running IPv6* . Apress, Nov. 2005.

[2] S. Frankel, R. Graveman, J. Pearce, M. Rooks. Guidelines for the Secure Deployment of IPv6. NIST SP 800-119, Dec. 2010.

[3] G. Huston. IPv4 Address Report. `http://www.potaroo.net/tools/ipv4/index.html`.

[4] kc claffy. Tracking IPv6 Evolution: Data We Have and Data We Need. *ACM SIGCOMM Computer Communication Review (CCR)*, (3):43–48, Jul 2011.

[5] D. Malone. Observations of IPv6 Addresses. In *Passive and Active Measurement Conference (PAM)*, pages 21–30, 2008.

[6] M. Leber. Global ipv6 deployment. `http://bgp.he.net/ipv6-progress-report.cgi`.

[7] M. Prior. IPv6 Status Survey. `http://www.mrp.net/IPv6_Survey.html`.

[8] T. Kuehne. Examining Actual State of IPv6 Deployment, 2008. `http://www.circleid.com/posts/81166_actual_state_ipv6_deployment`.

[9] E. Karpilovsky, A. Gerber, D. Pei, J. Rexford, A. Shaikh. Quantifying the Extent of IPv6 Deployment. In *Passive and Active Measurement Conference (PAM)*, pages 13–22, 2009.

[10] W. Shen, Y. Chen, Q. Zhang, Y. Chen, B. Deng, X. Li, G. Lv. Observations of IPv6 Traffic. In *ISECS Computing, Communication, Control, and Management (CCCM)*, pages 278–282, August 2009.

[11] C. Labovitz. IPv6 Momentum? `http://ddos.arbornetworks.com/2010/10/ipv6-momentum/`.

[12] S. Steffann. IPv6 test, 2008. `http://ipv6test.max.nl/`.

[13] L. Colitti, S. Gunderson, E. Kline, T. Refice. Evaluating IPv6 Adoption in the Internet. In *Passive and Active Measurement Conference*, pages 141–150. April 2010.

[14] R. Smets. IPv6 Deployment Monitoring, December 2010. `http://ipv6-ghent.fi-week.eu/files/2010/12/1335-Rob-Smets-v2.pdf`.

[15] E. Aben. Measuring World IPv6 Day - Long-Term Effects, 2011. `https://labs.ripe.net/Members/emileaben/measuring-world-ipv6-day-long-term-effects`.

[16] C. Huitema. Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380 (Proposed Standard), February 2006.

[17] D. Wing, A. Yourtchenko. Happy Eyeballs: Success with Dual-Stack Hosts. RFC 6555 (Proposed Standard), April 2012.

[18] Microsoft Technet. DNS Client Behavior in Windows Vista. `http://technet.microsoft.com/en-us/library/bb727035.aspx`.

[19] G. Huston, G. Michaelson. IPv6 Capability Tracker, 2012. `http://labs.apnic.net/tracker.shtml`.

[20] N. C. Zakas. How many users have JavaScript disabled?, Oct. 2010. `http://developer.yahoo.com/blogs/ydn/posts/2010/10/how-many-users-have-javascript-disabled/`.

[21] S. Ramachandran. Web metrics: Size and number of resources, 2010. `https://developers.google.com/speed/articles/web-metrics`.

[22] Browserscope. Community-driven project for profiling web browsers. `http://www.browserscope.org/?category=network`.

[23] MaxMind's GeoIP Country Database. `http://www.maxmind.com/app/geoip_country`.

[24] Cisco Systems. Visual Network Index. `http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html`.

[25] Wikimedia Traffic Analysis, March 2012. http://stats.wikimedia.org/wikimedia/squids/SquidReportPageViewsPerCountryOverview.htm.

[26] Advanced Network Technology Center, University of Oregon. University of Oregon Route Views Project, 2012. http://www.routeviews.org/.

[27] A. Korinek, J. A. Mistiaen, M. Ravallion. An Econometric Method of Correcting for Unit Nonresponse Bias in Surveys. *Journal of Econometrics*, 136(1):213–235, 2007.

[28] Wikipedia. Usage share of operating systems, May 2012. http://en.wikipedia.org/w/index.php?title=Usage_share_of_operating_systems&oldid=490169177.

[29] Wikipedia. Usage share of web browsers, May 2012. http://en.wikipedia.org/w/index.php?title=Usage_share_of_web_browsers&oldid=490374342.

[30] C. Kreibich, N. Weaver, B. Nechaev, V. Paxson. Netalyzr: Illuminating the Edge Network. In *ACM SIGCOMM Conference on Internet Measurement*, pages 246–259, 2010.

[31] R. Despres. IPv6 Rapid Deployment on IPv4 Infrastructures (6rd). RFC 5569 (Informational), January 2010.

[32] S. Zander, L. L. H. Andrew, G. Armitage, G. Huston, G. Michaelson. Investigating the IPv6 Teredo Tunnelling Capability and Performance of Internet Clients. *SIGCOMM CCR*. (accepted for publication, to appear).

[33] Chrome Repository. Chrome Happy Eyeballs. http://src.chromium.org/viewvc/chrome?view=rev\&revision=85934.

[34] E. Aben. Hampered Eyeballs. https://labs.ripe.net/Members/emileaben/hampered-eyeballs.

[35] Mozilla Bugzilla. Firefox happy Eyeballs. https://bugzilla.mozilla.org/show_bug.cgi?id=621558.

[36] M. Nikkhah, R. Guérin, Y. Lee, R. Woundy. Assessing IPv6 through web access a measurement study and its findings. In *Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, pages 26:1–26:12, 2011.

[37] CAIDA. IPv4 and IPv6 AS Core: Visualizing IPv4 and IPv6 Internet Topology at a Macroscopic Scale in 2010. http://www.caida.org/research/topology/as_core_network/.

[38] M. Defeche, E. Vyncke. Measuring IPv6 Traffic in BitTorrent Networks. IETF draft-vyncke-ipv6-traffic-in-p2p-networks-01.txt, March 2012.

[39] T. Anderson. IPv6 Dual-stack Client Loss in Norway. http://fud.no/ipv6/.