

Evaluating IPv6 Adoption in the Internet

Lorenzo Colitti, Steinar H. Gunderson, Erik Kline, Tiziana Refice

Google, Inc.

{lorenzo, sesse, ek, tiziana}@google.com

Abstract. As IPv4 address space approaches exhaustion, large networks are deploying IPv6 or preparing for deployment. However, there is little data available about the quantity and quality of IPv6 connectivity. We describe a methodology to measure IPv6 adoption from the perspective of a Web site operator and to evaluate the impact that adding IPv6 to a Web site will have on its users. We apply our methodology to the Google Web site and present results collected over the last year. Our data show that IPv6 adoption, while growing significantly, is still low, varies considerably by country, and is heavily influenced by a small number of large deployments. We find that native IPv6 latency is comparable to IPv4 and provide statistics on IPv6 transition mechanisms used.

1 Introduction

The network protocol that has been used in the Internet since its inception is IPv4 [1], which provides 2^{32} distinct addresses. Its successor IPv6 [2] provides 2^{128} addresses, but IPv6 adoption has not proceeded as quickly as its designers expected [3]. Since IPv6 is not backward-compatible with IPv4 both clients and servers have to deploy IPv6 to make use of it; since IPv6 provides little immediate benefit apart from larger address space the operational community has seen little motivation to deploy it. However, as IPv4 address exhaustion approaches [4], a number of networks have deployed IPv6 or are preparing for deployment.

One of the problems faced by an organization, especially a Web site operator, wanting to deploy IPv6 is the lack of information on IPv6 adoption and the quality of service provided by the IPv6 Internet. Thus, it is difficult to predict the adoption and impact to existing services of even a small-scale IPv6 roll-out. Conventional wisdom in the operational community is that low adoption leads to lower quality of service, since problems are less likely to be noticed, and lower quality of service leads to low adoption, as a network with lower quality of service is less desirable. However, little data is available to validate these assumptions. The situation is further complicated by the wide variety of mechanisms currently used to provide IPv6 connectivity. To work around an initial lack of IPv6 deployment, various transition mechanisms were developed to provide IPv6 connectivity on IPv4-only networks, often by encapsulating IPv6 traffic in IPv4. Examples are configured tunnels [5], 6rd [6] and ISATAP [7], which require operator-controlled intermediate nodes; and 6to4 [8] and Teredo [9], which can

use third-party relay nodes that may be anywhere on the Internet. An IETF survey [10] lists over 30 documents covering a multitude of scenarios.

Attempts have been made to quantify IPv6 adoption in various ways. Past work has counted IPv6 addresses observed at 6to4 relays [11, 12] or proxies [13], though this can only observe the users of the relay examined. Arbor Networks [14] sampled IPv6 traffic on backbones, but since current routers generally do not support IPv6 packet sampling, they could only observe tunneled traffic and were unable to conclude how much native traffic was present. Karpilovsky et al. [15] compared IPv6 address allocations published by Internet registries to BGP routing tables, finding that allocations were growing, but that many prefixes were used either long after allocation or not at all. They found that IPv6 traffic within an undisclosed tier-1 ISP was negligible, and mostly consisted of DNS and ICMP. However, tier-1 status in IPv4 does not necessarily imply a substantial IPv6 deployment; in fact, many IPv4 tier-1 ISPs currently only have a handful of IPv6 customers—for example, on 6 October 2009, AT&T (AS7018) only had four IPv6 BGP customers according to publicly-available BGP tables [16, 17]. They also looked at Teredo traffic, but as we note later, this is prone to undercounting since many implementations prefer IPv4 over Teredo. Zhou et al. [18] compared IPv6 and IPv4 one-way delay between 26 measurement points, finding that IPv6 latency exhibited larger variation, and was significantly higher for 36% of paths. Huston [3] analyzed logs from the RIPE and APNIC Web sites, finding that IPv6 connectivity was available to about 1% of their users. These Web sites are primarily targeted at network operators, who are likely to have more access to IPv6 connectivity than general Internet users. Kevin Day [19] used 1x1 pixel images on a Web page to track working IPv6, finding growth from 0.014% to 0.084% between December 2005 and March 2008, and broken IPv6, which varied between 0.2% and 0.4%. Wikimedia [20] conducted similar measurements between June 2008 to July 2009. A software package [21] is also available for Web site operators to measure working and broken IPv6 for their user base. However, to our knowledge there is no comprehensive study that examines IPv6 deployment from the perspective of a Web site operator or that can serve to quantify the impact of adding IPv6 support to a large Web site with a broad user base.

In this paper we present a methodology for characterizing IPv6 adoption, connectivity, and latency from the perspective of a Web site operator (Section 3). We apply the methodology to the Google Web site to conduct a large-scale study of IPv6 deployment. Our data (Section 4) helps answer questions such as: what percentage of users would use Google’s services over IPv6 if it were enabled? What would be the impact on reliability and latency? What is the extent of IPv6 deployment in various countries and networks, and which transition mechanisms are used?

2 Browser Behaviour

Predicting the impact of enabling IPv6 on a Web site requires an understanding of browser behaviour in a multi-protocol environment, as the availability of

two protocols creates a choice as to which protocol to use. Since IP addresses are inconvenient to remember, network applications such as Web browsers typically allow users to input names (e.g. `www.google.com`), and use the Domain Name System (DNS) to resolve names to IP addresses. The DNS uses A and AAAA *Resource Records* (RRs), respectively, to specify IPv4 addresses (e.g., 74.125.19.99), and IPv6 addresses (e.g., 2001:4860:b006::68). It also employs distributed caching of RRs based on the *Time-To-Live* (TTL) of each RR; when the TTL expires, a new RR must be requested. Web browsers supporting both IPv4 and IPv6 generally request both types of RRs; most recent operating systems allow limiting the requests to only those protocols that are available at query time. Browsers then typically attempt to connect to all the returned addresses in order, retrying with the next address if an attempt fails or times out [22]. The exact order is application and operating-system dependent [23], but virtually all IPv6-capable browsers will prefer native IPv6 to IPv4; some implementations will prefer IPv4 to tunneling mechanisms, some will not. Although this fallback behaviour can be used as a crude high-availability solution, content providers typically use *Virtual IP addresses* (VIPs), backed by multiple Web servers, instead of relying on browser behaviour.

3 Measurement Methodology

Our measurement methodology is based on asking Web clients to send HTTP requests to either an IPv4-only host or a dual-stack host and comparing the results. For this purpose, we use two hostnames (the *ExpHostNames*):

- `dualstack.ipv6-exp.1.google.com`: a dual-stack hostname, i.e. a hostname with both AAAA and A DNS RRs, (the ExpHostName_D), and
- `ipv4.ipv6-exp.1.google.com`: an IPv4-only hostname, i.e. a hostname with an A DNS RR only (the ExpHostName_4).

The *ExpHostNames* correspond to a set of VIPs (the *ExpVIPs*), which are load-balanced among a set of Web Servers, (the *ExpWSes*) in geographically distinct datacenters. Each datacenter has a pair of *ExpVIPs*, one IPv4 and one IPv6. To allow comparison of IPv4 and IPv6 results, we configure the Google DNS servers so that DNS requests from a given resolver (and thus from a given client) will return *ExpVIPs* in the same datacenter. We set a low TTL value (5 seconds) on the DNS records for both *ExpHostNames*. This ensures that most requests require DNS lookups, minimizing latency differences due to caching. In September 2008, we set up *ExpVIPs* in two datacenters (one in the US, one in Europe). Between March 2009 and June 2009, we added three more (one in the US, one in Europe, and one in Asia). In June 2009, the Asian *ExpVIPs* were turned down for unrelated reasons. Since then we have been using four datacenters.

To cause Web clients to connect to the *ExpVIPs*, we modify the results returned for a small, randomly-selected fraction of Google search requests, which we name *SearchRequests*. Fig. 1 describes the measurement process. Specifically, when a *SearchRequest* is processed by a Web Server (the *SearchWS*):

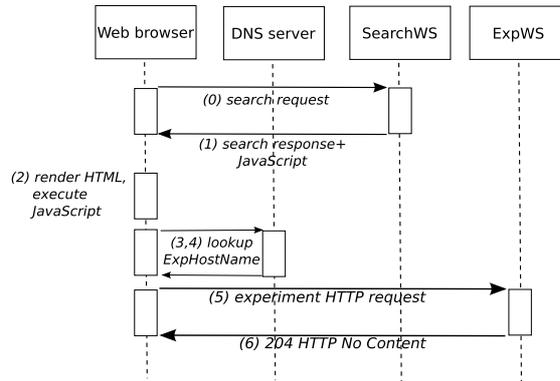


Fig. 1. Measurement Request Flow

1. SearchWS adds to the search results a JavaScript fragment which instructs the browser to fetch a URL (the *ExpURL*) containing an ExpHostName.
2. If the browser executes the JavaScript code, it will:
 - (a) Execute a DNS lookup for the ExpHostName (either A or AAAA+A)
 - (b) Receive an IPv4 and/or an IPv6 ExpVIP (depending on which ExpHostName and which DNS records were requested)
 - (c) Send an HTTP request (the *ExpRequest*) to the ExpVIP
3. If ExpWS receives the ExpRequest, it logs it and returns to the browser a HTTP 204 No Content response.

The URL loaded by the JavaScript contains the following elements:

ExpHostName 10% of the SearchRequests are sent to ExpHostName₄, while 90% are sent to ExpHostName_D. We arbitrarily chose the 90/10 split because we expected the vast majority of hosts to connect over IPv4, and wanted to collect as much IPv6 data as possible.

SearchClientIP The IPv4 or IPv6 address of the client, as seen by SearchWS.

SearchTS The timestamp of the SearchRequest, as measured by SearchWS.

HashCode A cryptographic hash of the other elements which can be used to verify that the ExpRequest is genuine.

The ExpURL is constructed as follows:

`http://ExpHostName/gen_204?ip=SearchClientIP&ts=SearchTS&auth=HashCode`

4 Data Analysis

Data collection started in September 2008 and is still ongoing. In this paper we present data collected up to September 2009. Each ExpRequest logged by an ExpWS is termed a *hit*. We receive order of millions of hits per day. Hits received on ExpHostName_D are termed *dual-stack hits*; of these, hits received over IPv6

are termed *IPv6 hits*. In every hit, we examine the following fields in addition to the fields in the ExpURL:

ExpClientIP The IP address which sent the ExpRequest.

ExpTS The timestamp of when ExpWS processed the ExpRequest.

UserAgent The User-Agent header in the ExpRequest; identifies the browser.

Logs are processed once per day, using MapReduce [24] and Sawzall [25] for efficiency. Before performing data analysis, we exclude hits from Google’s own networks as well as invalid and duplicate hits, as follows. First, in order to reduce the likelihood of users altering their or others’ requests (e.g., by arbitrarily delaying IPv6 requests, or by sending IPv6 and IPv4 requests through different hosts), we exclude all hits with an invalid HashCode. The number of such hits is negligible. Second, to avoid duplicate hits due, for example, to replay attacks or users clicking on the browser’s back button and causing a second ExpRequest to be issued, we exclude all but the first hit with a given HashCode in a given day. To avoid duplicates across days, we discard all hits which are logged more than $MaxResponseTime = 5$ minutes after the corresponding SearchRequests. If a request does not succeed within 5 minutes, we consider that client not to have working IPv6, as we consider this is not an acceptable response time for a Web page. This does not significantly bias our data, because – as can be seen in Fig. 8 – about 94% of all hits are received in the first 3.5 seconds.

By analyzing the hits in our dataset, we infer various statistics such as the clients that can or cannot connect to dual-stack Web servers and their request latency. The remainder of this section describes these statistics.

IPv6 Connectivity Ratio To measure the availability of IPv6 connectivity among Google users, we count the number of IPv6 hits. Every IPv6 hit implies that the client that sent it has *working IPv6*. A SearchRequest will not result in a hit in at least the following cases:

1. The browser has JavaScript disabled or does not accept third-party images.
2. The user navigates away from the result page, loses connectivity, or closes the browser before the ExpURL has been loaded.
3. The browser attempts to contact the dual-stack host $ExpHostName_D$, but cannot reach it within $MaxResponseTime$. In this case, we say that the client has *broken IPv6*. Since our measurement methodology does not allow us to distinguish this case from the others, we do not present data on broken IPv6.

We define the *working IPv6 ratio* W as the number of IPv6 hits divided by the total number of hits to the dual-stack host: $W = \frac{n_{D6}}{n_{D4} + n_{D6}} = \frac{n_{D6}}{n_D}$, where n_{D4} and n_{D6} are, respectively, the number of IPv4 and IPv6 hits to the dual-stack host and n_D is their total. Since W is computed by sampling a constant value (the percentage of IPv6 connectivity) using a fixed number of independent trials, we treat it as being binomially distributed. Also, as shown by Fig. 2, n_D is very large compared to n_{D6} . Thus, we can approximate the binomial

as a normal distribution and calculate the standard deviation of W as: $\sigma_W = \sqrt{W(1-W)/n_D}$.

As an example, during the week of 2009-09-20, about 0.252% of all dual-stack hits were IPv6, and $\sigma_W = 0.001\%$. σ_W is of the same order of magnitude for every week in our dataset. In the following, we use weekly numbers (i.e. we aggregate all the hits for one week) unless otherwise noted. If we assume the average number of searches per day made by a given user does not depend on whether the user has IPv6 connectivity or not, we can use W as an estimator of the ratio of Google users with working IPv6.

Connectivity Over Time Fig. 2 plots the value of W over time since we started collecting data. Although the percentage is still low, it has grown significantly in the last 12 months. For example, for the week of 14 September, year-over-year growth is approximately 35%. We attribute the dip in IPv6 connectivity in August to seasonal variations: since – as we see in Fig. 6 – IPv6 deployment is heavily influenced by a small number of countries, IPv6 traffic is more prone to the effects of holiday seasons than IPv4 traffic. W varies during the course of a week and is substantially higher during the weekends. An example is Fig. 3, which shows September 2009. This suggests that IPv6 is more available to users at home than in their workplace.

Connectivity by Type To infer the type of IPv6 connectivity used by clients, we examine the ExpClientIP address of each IPv6 hit, as described in [13, 15]. This allows us to distinguish 6to4, Teredo, and ISATAP hits. Other hits, which include both native traffic and other transition mechanisms such as configured tunnels, cannot be distinguished based on IP address alone. As shown in Fig. 4, 6to4 is the most common connectivity type, while ISATAP and Teredo are comparatively rare. Note that this is not an indication of what type of connectivity is available to users, but what type of connectivity would be used to connect to dual-stack Web sites. For example, implementations such as Windows Vista prefer IPv4 over Teredo and 6to4, and thus we will not observe them (Section 2).

Connectivity by Operating System We infer the operating system of the client based on the HTTP User-Agent header of each IPv6 hit. We only take into account OSes that, during September 2009, had a significant number of IPv6 hits (i.e. $W^{os} \geq 1\%$ of all IPv6 hits). As shown in Fig. 5, most IPv6 hits are from MacOS and Windows Vista clients. This highlights the importance of operating system defaults. In fact, the number of IPv6 hits from Windows XP (which is not IPv6-enabled by default) is approximately three times lower than those from Windows Vista (which is IPv6-enabled by default), even though Windows XP's market share is approximately three times higher [26]. Further analysis of the data shows that most MacOS IPv6 hits use 6to4 ($\simeq 90\%$, while all the other OSes have $< 50\%$). We do not know the reason for this. We are unable to infer the operating system for 0.082% of IPv6 hits.

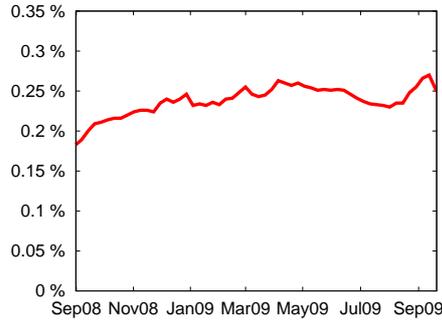


Fig. 2. Working IPv6 over time.

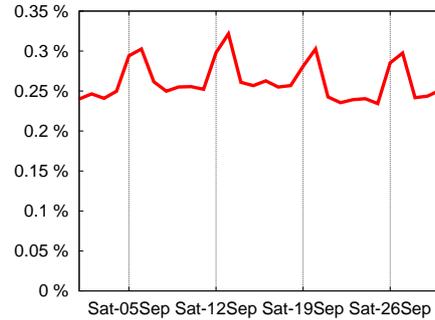


Fig. 3. Daily working IPv6 in Sep 2009.

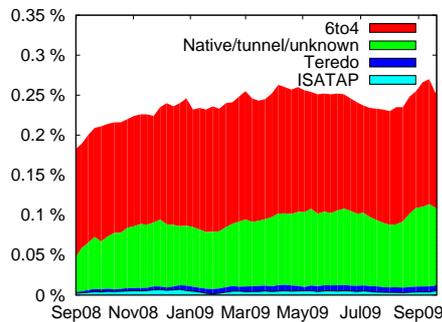


Fig. 4. Working IPv6 by connectivity type.

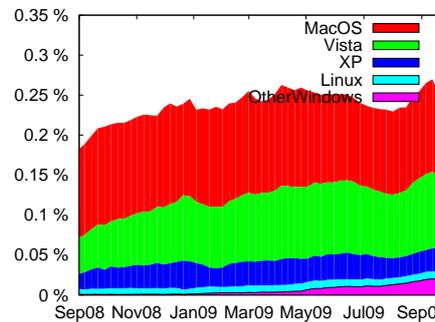


Fig. 5. Working IPv6 by operating system.

Connectivity by Country To determine IPv6 availability and connectivity types in different countries, we geolocate the ExpClientIP using internal geolocation databases. We then consider the ratio $W_{ct}^{cc} = n_{6ct}^{cc} / n_D^{cc}$ between IPv6 hits using connectivity type ct from country cc and all dual-stack hits from cc . We only take into account countries accounting for at least 1% of all IPv6 hits in September 2009. As shown in Fig. 6, there are both significant differences in IPv6 adoption (almost an order of magnitude) and in connectivity types between countries. We note that availability of IPv6 connectivity in a given country does not necessarily correlate with IPv6 deployment, since relayed transition mechanisms such as 6to4 and Teredo can be enabled by users in the absence of IPv6 network infrastructure. A better measure of the deployment of IPv6 in a given country can be obtained by removing relay mechanisms. Fig. 7 shows that the most significant deployments of IPv6 are in France and China. The high proportion of 6to4 in Russia and Ukraine may be due to the Opera browser, which prefers 6to4 over IPv4; our internal data shows that it is popular in those countries.

Connectivity by AS We then determine the Autonomous Systems which originate IPv6 hits by looking up the ExpClientIP in the BGP routing tables

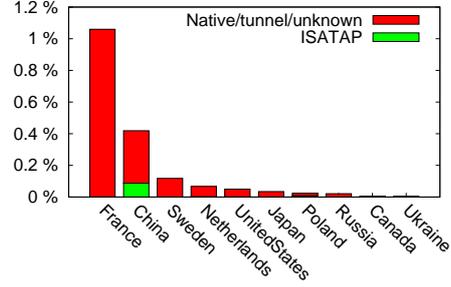
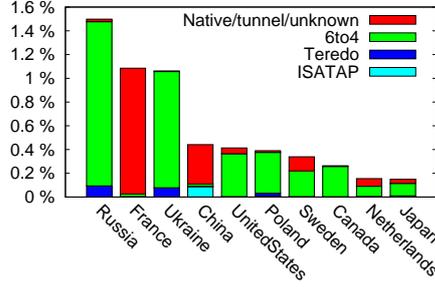


Fig. 6. Working IPv6 ratio for top-10 countries by connectivity type. **Fig. 7.** Working IPv6 ratio for top-10 countries, non-relayed only.

of Google’s routers. Then, for each AS, we compute the working IPv6 ratio $W^{AS} = n_{D6}^{AS}/n_D^{AS}$, where n_{D6}^{AS} and n_D^{AS} are, respectively, the number of dual-stack hits and IPv6 hits (excluding 6to4 and Teredo) coming from IP addresses originated by that AS. Table 1 presents the top ASes with higher W^{AS} (during September 2009). We filter out ASes that contribute only marginally to the experiment, i.e. with $n_{D6}^{AS} < 1\%$ of all the IPv6 hits in the same period. Only 7 ASes match this criterion. Moreover, in order to identify ASes with higher impact on overall IPv6 traffic, we also compute $W_{D6}^{AS} = n_{D6}^{AS}/n_{D6}$, where n_{D6} is the total number of IPv6 hits (excluding 6to4 and Teredo) in September 2009. Table 2 shows the top ASes with higher W_{D6}^{AS} (during September 2009). Note the ASes in both tables are the same. 6 out of 7 ASes shown in Table 1 and 2 are universities or research institutions. The only exception - Free (AS12322) - contributes to most of the French IPv6 native hits (presented in Fig. 7).

Table 1. IPv6 connectivity per origin AS vs dual-stack connectivity for the same AS. **Table 2.** IPv6 connectivity per origin AS versus overall IPv6 connectivity.

ASN	AS Name	Country	W^{AS}	ASN	AS name	Country	W_{D6}^{AS}
37944	CSTNET	CN	100.000%	12322	Free	FR	54.956%
23910	CERNET2	CN	99.962%	23910	CERNET2	CN	7.160%
19782	Indiana Uni.	US	89.325%	1312	Virginia Tech	US	2.472%
1312	Virginia Tech	US	51.743%	4538	CERNET-BKB	CN	2.001%
6122	ICN	US	13.449%	19782	Indiana Uni.	US	1.428%
12322	Free	FR	5.131%	6122	ICN	US	1.178%
4538	CERNET-BKB	CN	2.650%	37944	CSTNET	CN	1.038%

IPv6 vs IPv4 Latency In order to compare the latency of HTTP queries performed over IPv4 and IPv6, we consider the latency of an ExpRequest (the

ExpLatency) as $\Delta t = \text{ExpTS} - \text{SearchTS}$ and we aggregate hits into 50 ms buckets. Fig. 8 plots latency computed since 18th June 2009. The graphs in Fig. 8 account for approximately 94% of all hits (the rest are received after the 3.5 second cut-off). As we can see, ExpRequests sent to ExpHostName₄ and to ExpHostName_D on IPv4 have almost identical latency. Either these clients did not attempt to request an IPv6 record or any added latency of doing so was insignificant. Thus, assigning a Web site an IPv6 address in addition to an IPv4 address did not affect the latency of IPv4-only clients in any measureable way.

We also see that *IPv6 latency* (i.e. latency of requests to ExpHostName_D over IPv6) is in general slightly higher than IPv4 latency. However, if we exclude relayed IPv6 connectivity (i.e., 6to4 and Teredo), IPv6 latency is actually lower than IPv4 latency. This is likely due to the fact that, as shown in Table 2, IPv6 deployment is heavily dominated by research and education networks and one large broadband ISP. We would expect these networks to have higher-bandwidth, lower-latency connections than average.

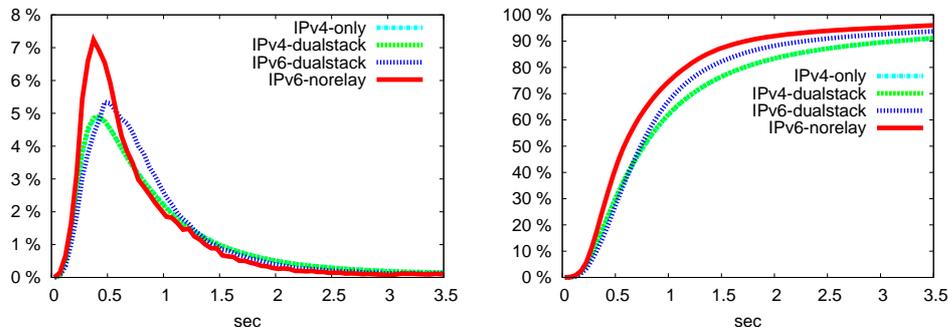


Fig. 8. PDF and CDF of hit latency. Time granularity of 50 ms. The IPv4-only and IPv4-dualstack plots are indistinguishable. The latency data are not indicative of ordinary Google service latency.

5 Conclusions and Future Work

We have presented methodologies that allow us to characterize several aspects of IPv6 adoption, connectivity, and quality. We have applied them to a large data set, and shown that IPv6 deployment is small but growing steadily, and that adoption is still heavily influenced by a small number of large deployments. While we see IPv6 adoption in research and education networks, IPv6 deployment is, with one notable exception, largely absent from consumer access networks. We find that native IPv6 latency is not significantly worse than IPv4, but transition mechanisms such as 6to4 and Teredo can add noticeable latency, perhaps because relays can be very far away from the users they serve. Finally, we have shown

that the default settings of operating systems and applications factor strongly in the level of IPv6 adoption seen on those platforms.

We believe that our methodology usefully characterizes properties of connectivity in the IPv6 Internet and intend to continue our measurements to provide a baseline as adoption grows. We also plan to conduct further measurements to quantify the incidence of broken IPv6 and its causes.

References

1. Information Sciences Institute: Internet Protocol. RFC 791 (1981)
2. Deering, S., Hinden, R.: Internet Protocol, Version 6 (IPv6). RFC 2460 (1998)
3. Huston, G.: IPv6 Transition. <http://www.potaroo.net/presentations/2009-09-01-ipv6-transition.pdf>
4. Huston, G.: IPv4 Address Report. <http://www.potaroo.net/tools/ipv4/>
5. Nordmark, E., Gilligan, R.: Basic Transition Mechanisms for IPv6 Hosts and Routers. RFC 4213 (2005)
6. Després, R.: IPv6 Rapid Deployment on IPv4 infrastructures (6rd). <http://tools.ietf.org/html/draft-despres-6rd>
7. Templin, F., Gleeson, T., Thaler, D.: Intra-Site Automatic Tunnel Addressing Protocol (ISATAP). RFC 5214 (2008)
8. Carpenter, B., Moore, K.: Connection of IPv6 Domains via IPv4 Clouds. RFC 3056 (2001)
9. Huitema, C.: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380 (2006)
10. IPv6 Operations Working Group charter. <http://ietf.org/dyn/wg/charter/v6ops-charter.html>
11. Savola, P.: Observations of IPv6 Traffic on a 6to4 Relay. SIGCOMM CCR (2005)
12. Hei, Y., Yamazaki, K.: Traffic Analysis and Worldwide Operation of Open 6to4 Relays for IPv6 Deployment. Symposium on Applications and the Internet (2004)
13. Malone, D.: Observations of IPv6 Addresses. In: PAM. (2008)
14. Arbor Networks: Tracking the IPv6 migration. Technical report (2008)
15. Karpilovsky, E., Gerber, A., Pei, D., Rexford, J., Shaikh, A.: Quantifying the Extent of IPv6 Deployment. In: PAM. (2009)
16. RIPE NCC: Routing Information Service. <http://www.ris.ripe.net/>
17. Hurricane Electric: IPv6 BGP table. <http://ipv6.he.net/bgpview/bgp-table-snapshot.txt>
18. Zhou, X., Mieghem, P.V.: Hopcount and E2E Delay: IPv6 Versus IPv4. In: PAM. (2005)
19. Day, K.: Working vs. Broken IPv6 Clients. <http://your.org/v6clients.png> (2008)
20. Wikimedia: IPv6 Deployment Status. http://wikitech.wikimedia.org/view/IPv6_deployment
21. Ward, N.: IPv6 WWW Test. <http://www.braintrust.co.nz/ipv6wwwtest/>
22. Hagino, J.: Implementing AF-Independent Application. <http://www.kame.net/newsletter/19980604/> (1998)
23. Draves, R.: Default Address Selection for Internet Protocol version 6 (IPv6). RFC 3484 (2003)
24. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI. (2004)
25. Pike, R., Dorward, S., Griesemer, R., Quinlan, S.: Interpreting the Data: Parallel Analysis with Sawzall. Scientific Programming Journal (2005)
26. Wikipedia: Usage Share of Desktop Operating Systems Retrieved on 2009-10-02.