Measuring Internet Client IPv6 Capabilities

Sebastian Zander¹, Lachlan Andrew¹, Grenville Armitage¹, Geoff Huston², George Michaelson²

szander@swin.edu.au

1 Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology
2 Asia Pacific Network Information Centre (APNIC)

Overview

- IPv4 address shortage and IPv6
- Investigate client’s IPv6 readiness
- Sample error mitigation
- Measured IPv6 capabilities
- Future Work
IPv4 Address Shortage and IPv6

IPv4 address space run out

- 4,294,967,296 ($2^{32}$) addresses BUT
  - Private, loopback, multicast, experimental (>13%)
  - Legacy A, B, C classes (before CIDR)
  - Hierarchical routing / sparse allocation
- Everything is connected now
  - Computers, phones, PVRs, fridges, houses, …
- Run out of unallocated IPv4
  - How much is actually used? → Another talk
Unusable IPv4 address space

IPv4 address space run out

- Geoff’s model (http://www.potaroo.net/tools/ipv4)
IPv6 solves address shortage …

- IPv6 addresses are **128 bit**
- 340,282,366,920,938,463,463,374,607,431,768,211,456 (3.4∙10^{38}) addresses
- New IPv6 features: integrated mobility, integrated multicast, auto-configuration, privacy, …
- Not backwards compatible → transition technologies
  - Dual stack systems
  - IPv6 over IPv4 tunneling (ISATAP, 6to4, 6rd, Teredo, …)
  - IPv4 over IPv6 tunneling (4over6, …)

… BUT nobody wants it

- IPv6 standardised 1998, still not widely used
  - Not backwards compatible
  - No compelling new features
  - Security concerns
  - Bad dual stack / tunneling performance
  - Network Address Translation (NAT)
- Let’s wait and see approach
- However, IPv6 fully supported by major OS and backbone providers, some movement among ISPs
How IPv6-ready is the Internet?

- IPv6 day
- Test web/email/DNS servers
- Study allocated/advertised BGP prefixes
- Traffic trace analysis
- Test client capabilities
- Research overview

IPv6 over IPv4 tunneling: 6to4

- Prefix 2002::/16
- IPv4 address encoded in IPv6 address
- Translation by relays
- Relays must have public IPv4 address
- 6to4 uses protocol type 41 (often filtered by firewalls)
IPv6 over IPv4 tunneling: Teredo

- Prefix 2001:0::/32
- Teredo can tunnel IPv6 over IPv4 through NATs
  - Teredo client learns its public IP address
  - NAT/firewall hole punching (modified STUN protocol)
  - Same relay handles both directions
- Present on Windows Vista and 7, but by default both do not send AAAA queries with only Teredo → IPv6 but no name resolution
Data Collection Setup

Client testing experimental setup

- Test script embedded in web pages
  - Script loads few invisible images and measures fetch latency (fetch by browser in background) → sub-tests
  - Image URL resolving depending on sub-test
  - Different server IP addresses for different sub-tests → browser must open new connection
  - Send summary message (with latencies) when all images loaded or after 10 second timeout

- Negligible impact on user
  - Test after page loads, small amount of data
Client testing experimental setup

- Test DNS, web server, local Teredo/6to4 relays
- Multiple of these for load-balancing
- Log DNS/HTTP requests, tcpdump traffic

![Diagram showing the setup with a client, internet, LAN, and DNS/HTTP requests logging.]

Test if client can do IPv4

- Client sends a query to the DNS server: `ID.v4only.server.domain`
- A Record: `192.168.4.23`
- Fetch Delay
- GET `192.168.4.23/1x1.png?ID.v4only`
- Can use IPv4
Test if client can do IPv6

Client

Query: ID.v6only.server.domain

AAAA Record: 2001:0DB8::23

GET 2001:0DB8::23/1x1.png?ID.v6only

Can use IPv6

Test dual stack preference

Client

Query: ID.dual.server.domain

A Record: 192.168.0.23
AAAA Record: 2001:0DB8::23

GET 2001:0DB8::23/1x1.png?ID.dual

Prefers IPv6
Test dual stack preference

Client

Query: ID.dual.server.domain
A Record: 192.168.0.23
AAAA Record: 2001:0DB8::23

Web Server

Get 192.168.0.23/1x1.png?ID.dual

Prefers IPv4

Test latent Teredo capability

Client

Get 2001:0DB8::23/1x1.png?ID.v6literal

DNS Server

Web Server

Can be forced to use IPv6
How are the clients selected?

- Participating sites link Javascript test (**JS-test**)
  - Sites can choose to turn on/off test
  - Script reports via Google Analytics
  - “All” visiting clients are tested
  - Cookies limit tests to one per host per day

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Test Events</th>
<th>Unique Events</th>
<th>Good Value</th>
<th>Avg Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>44</td>
<td>44</td>
<td>37.86</td>
<td>37.86</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>46</td>
<td>46</td>
<td>18.66</td>
<td>18.66</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>40</td>
<td>40</td>
<td>31.36</td>
<td>31.36</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>39</td>
<td>39</td>
<td>17.09</td>
<td>17.09</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>31</td>
<td>31</td>
<td>9.29</td>
<td>9.29</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>20</td>
<td>20</td>
<td>15.79</td>
<td>15.79</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>16</td>
<td>16</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

How are the clients selected?

- Test script inside Flash Ad banner (**FA-test**)
  - Distributed to clients via Google AdSense
  - Test executed when Ad is **displayed**
  - Google controls ad distribution
  - Only clients with Flash can be tested

Are You **IPv6** Ready?
Sample Error Mitigation

Sample error mitigation

- We only have IPv4 + browser ID
- Per-client statistics may be biased
  - Changing IP addresses (DHCP, mobility)
  - Sharing IP addresses (proxies, NATs)
  - More likely to see “heavy users”, which are potentially are more up-to-date (with IPv6)
- Interpret results as statistics of connections
Sample error mitigation

- Test “random” clients, but not uniform random
  - Certain web sites (JS-test)
  - Google’s ad distribution (FA-test)
- Check for bias by comparing with “known” population characteristics: OS and browser distribution
- Mitigate bias by reweighting each test by client countries traffic proportion
  - Cisco (ISP monitoring), Wikipedia

Browser proportions

![Browser proportions chart](chart.png)
OS proportions

Windows version proportions
Completed tests per day

Coverage /24 Subnets
Client IPv6 Capabilities

Results: IPv6 forced, capable, preferred

---

![Graph showing IPv6 connections over time.](image-url)
Results: IPv6 capable by type

Results: IPv6 preferred by type
Results: IPv6 capable by OS

Results: IPv6 preferred by OS
Results: Windows IPv6 capable

Results: Windows IPv6 preferred
Results: Happy eyeballs (fast failover)

Results: Happy eyeballs by browser
Results: IPv6 added fetch latency

Fetch latency difference of IPv6 and IPv4 [s]

Future Work and Summary
Future Work

- Increase sample size
- Estimate sample error
- Improve sampling error mitigation
- Hopefully more trends

Summary

- 6—7% IPv6 capable (but only 1—2% native)
- 1—2% prefer IPv6 (only the natives)
- 15—16% additional capable with unconstrained Teredo
- Teredo adds 1—2 seconds delay with optimally located relay (that’s why it’s constrained)
- Windows, MacOS X & Linux “equally” capable, but Windows fewer preferred (relies more on 6to4)
- Happy eyeballs on the rise
Acknowledgements

- 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.

Questions?