

IPv6 Measurements from Cisco's 6lab

Pierre-Alain Dupont and Nicolas looss

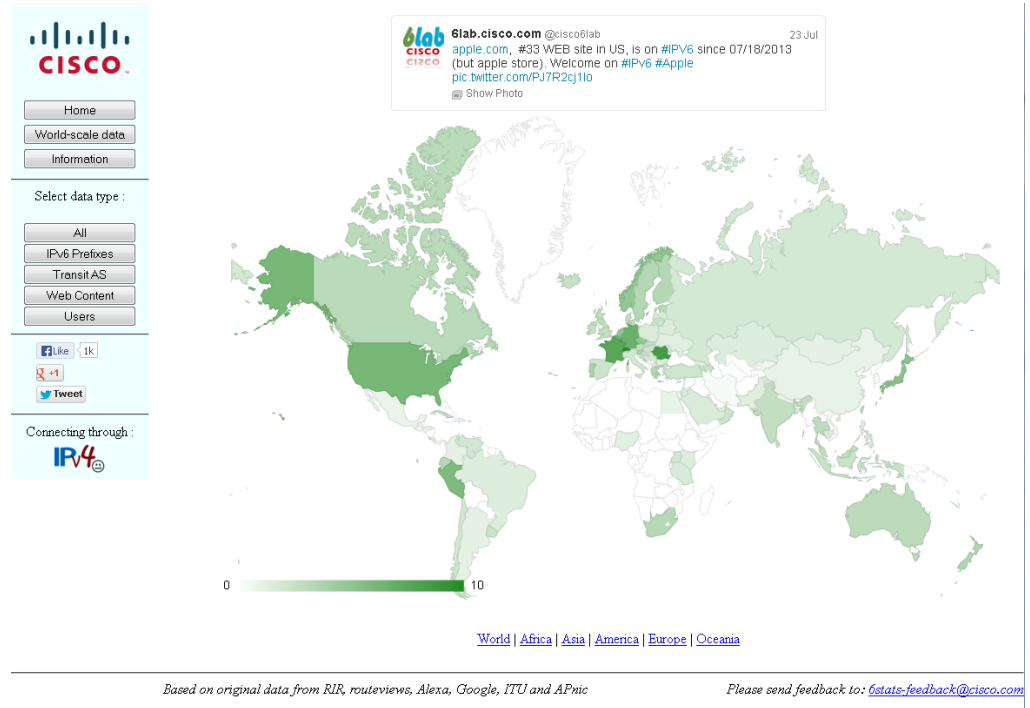
Cisco interns

Content

1. Current IPv6 measurements
2. APNIC experiment replication
3. Unexpected results

<http://6lab.cisco.com/stats>

- Available content
- Infrastructure
- Users



Improve user data

Which data is needed?

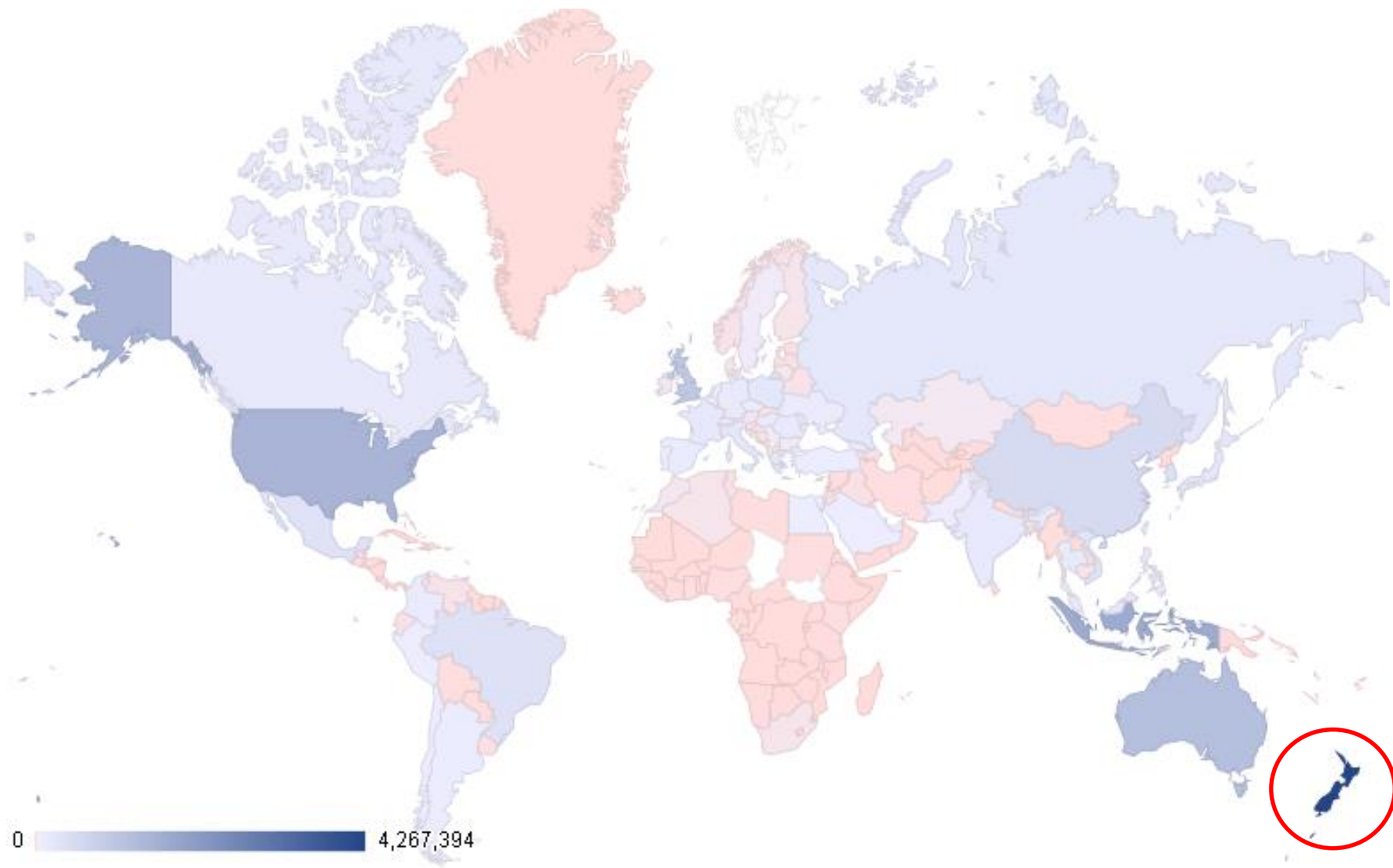
- Collect IPv6 adoption ratio.
- Representative distribution of users among ISPs/countries.
- Aggregated on a per-ISP basis.

Most publicly-available data:

- aggregated by country
- or only target few ASes.

So we need to build a system to gather such data!

APNIC experiment: user distribution



APNIC experiment

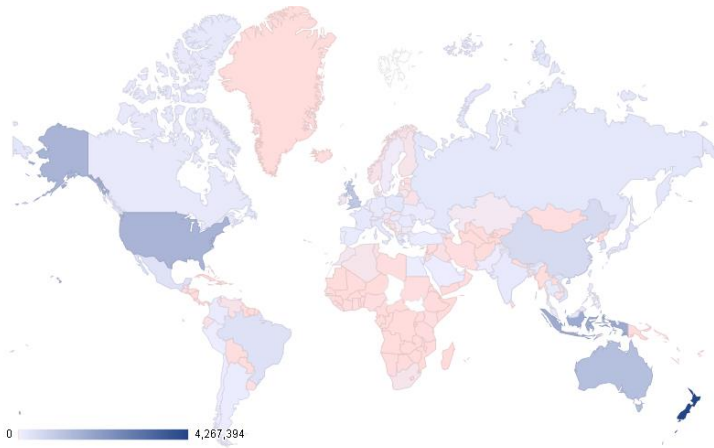
How to represent the bias of the geographical distribution ?

A ratio between experiments and potential users

We use a logarithm for a more symmetric scale

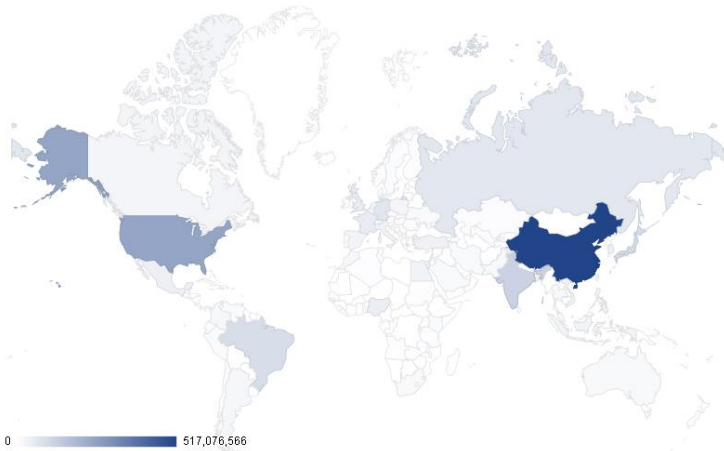
- Otherwise the ratio ranges from either $[0,1]$ or $[1, +\infty[$
- The logarithm is centered on 0 and treats those two cases equally

$$\text{Final formula : } \log \frac{\#experiences_{country} / \#users_{country}}{\#experiences_{world} / \#users_{world}}$$

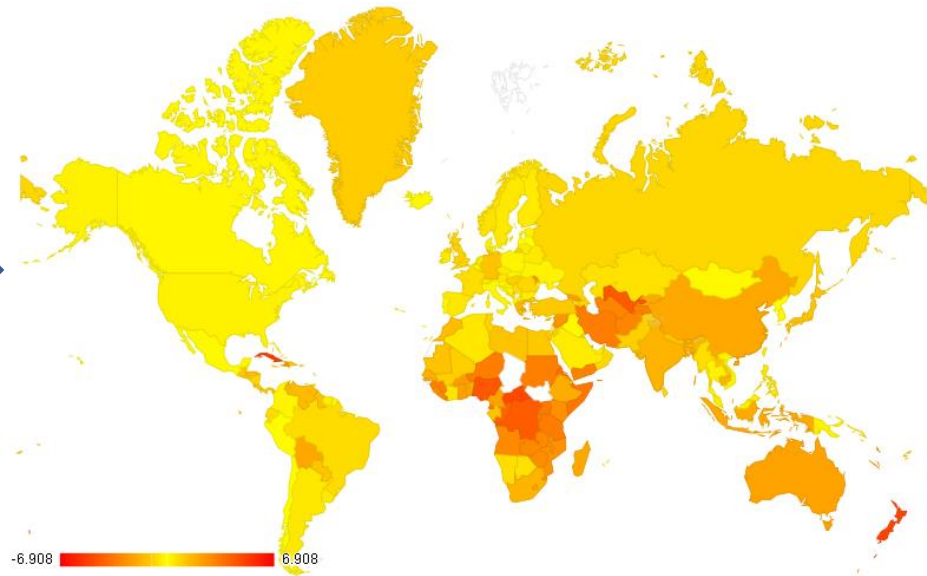


Number of experiments (APNIC)

Number of Internet users (ITU)



$$\log \left(\frac{\frac{\#experiments_{country}}{\#users_{country}}}{\frac{\#experiments_{world}}{\#users_{world}}} \right)$$



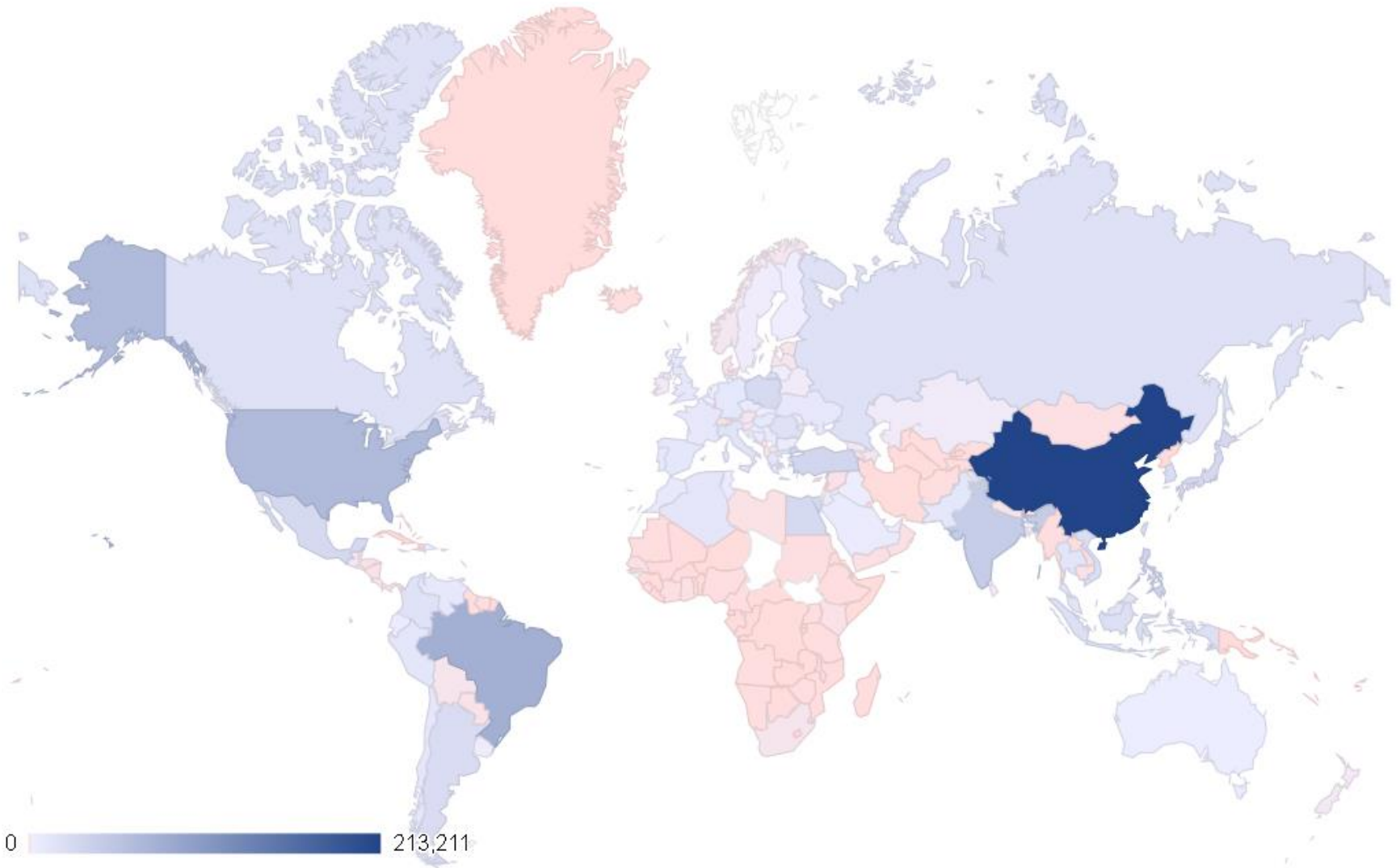
Comparative map

Replicate the experiment

Slight changes:

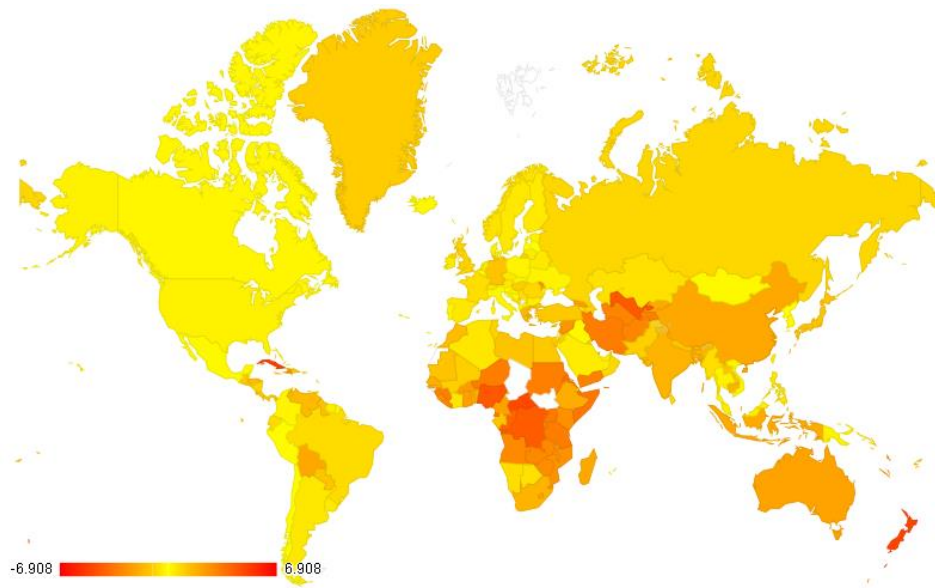
- Account in UTC+2
- Smaller server (1GB Linode) in the UK
- Don't use keywords but categories
- Use a dedicated IPv6 address (in one /64 prefix) for each experiment. Use CPM rather than CPC
- Tried Yahoo! ads, but too expensive (\$500/day)

A different geographical distribution

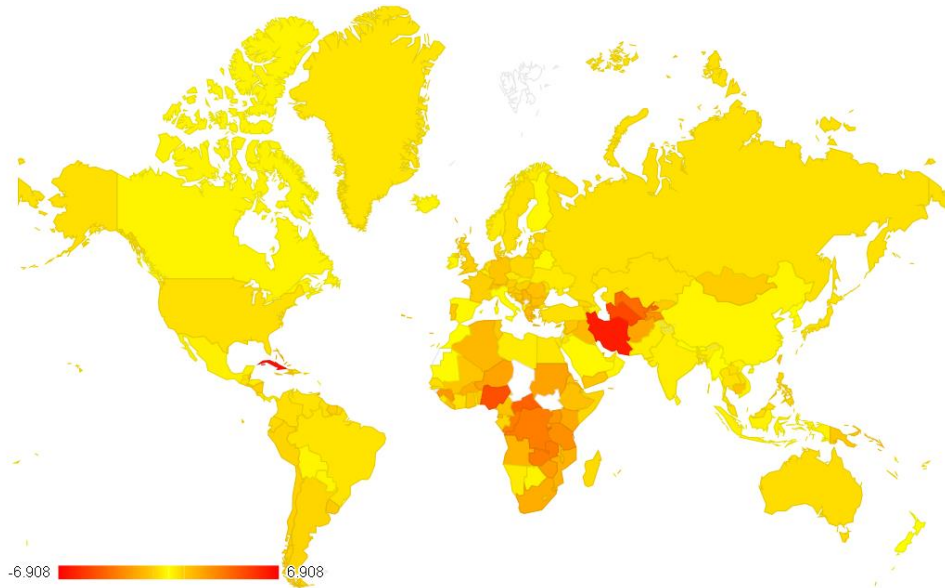


A different geographical distribution

$$\log \left(\frac{\frac{\#experiments_{country}}{\#users_{country}}}{\frac{\#experiments_{world}}{\#users_{world}}} \right)$$

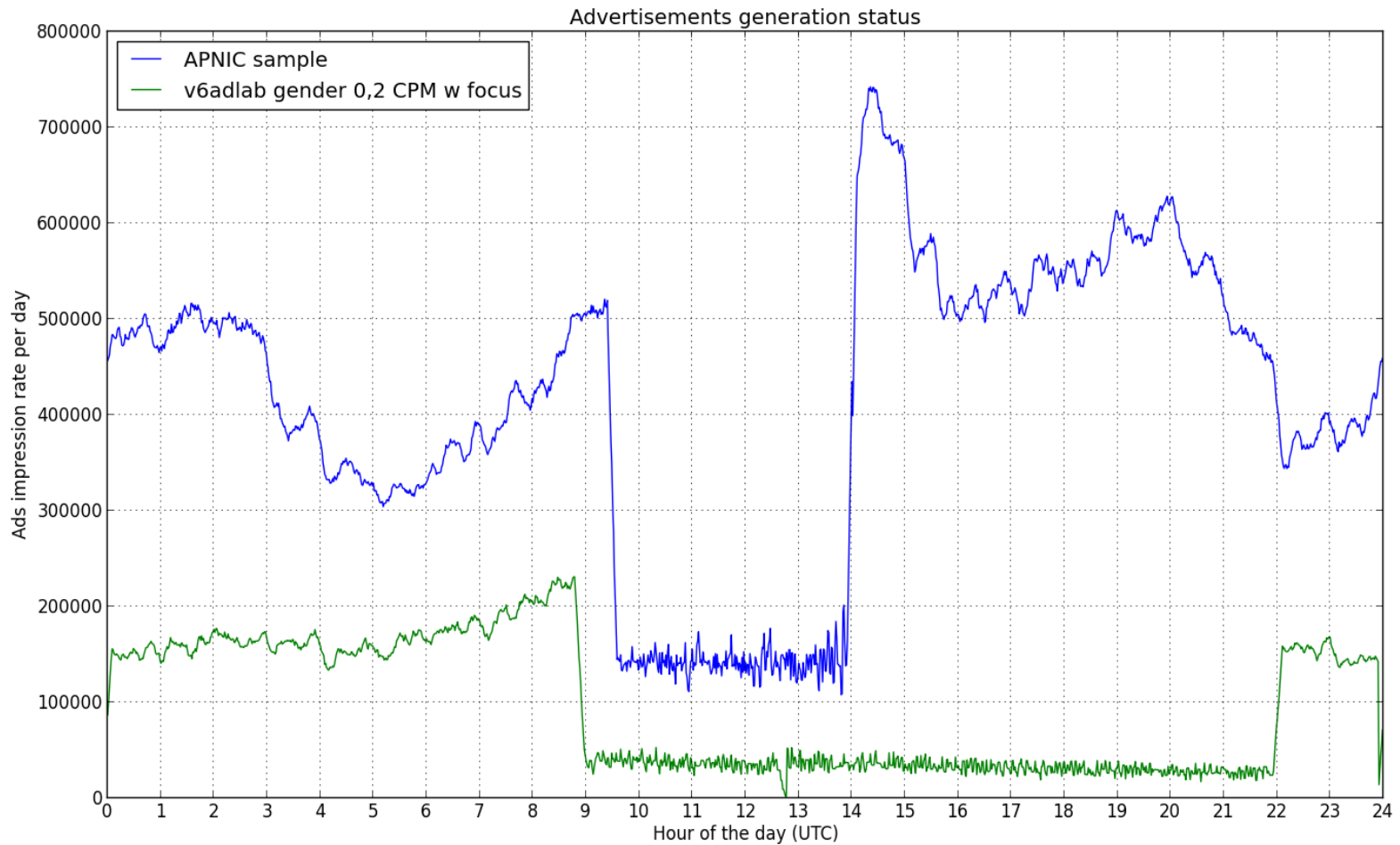


Space bias (APNIC)

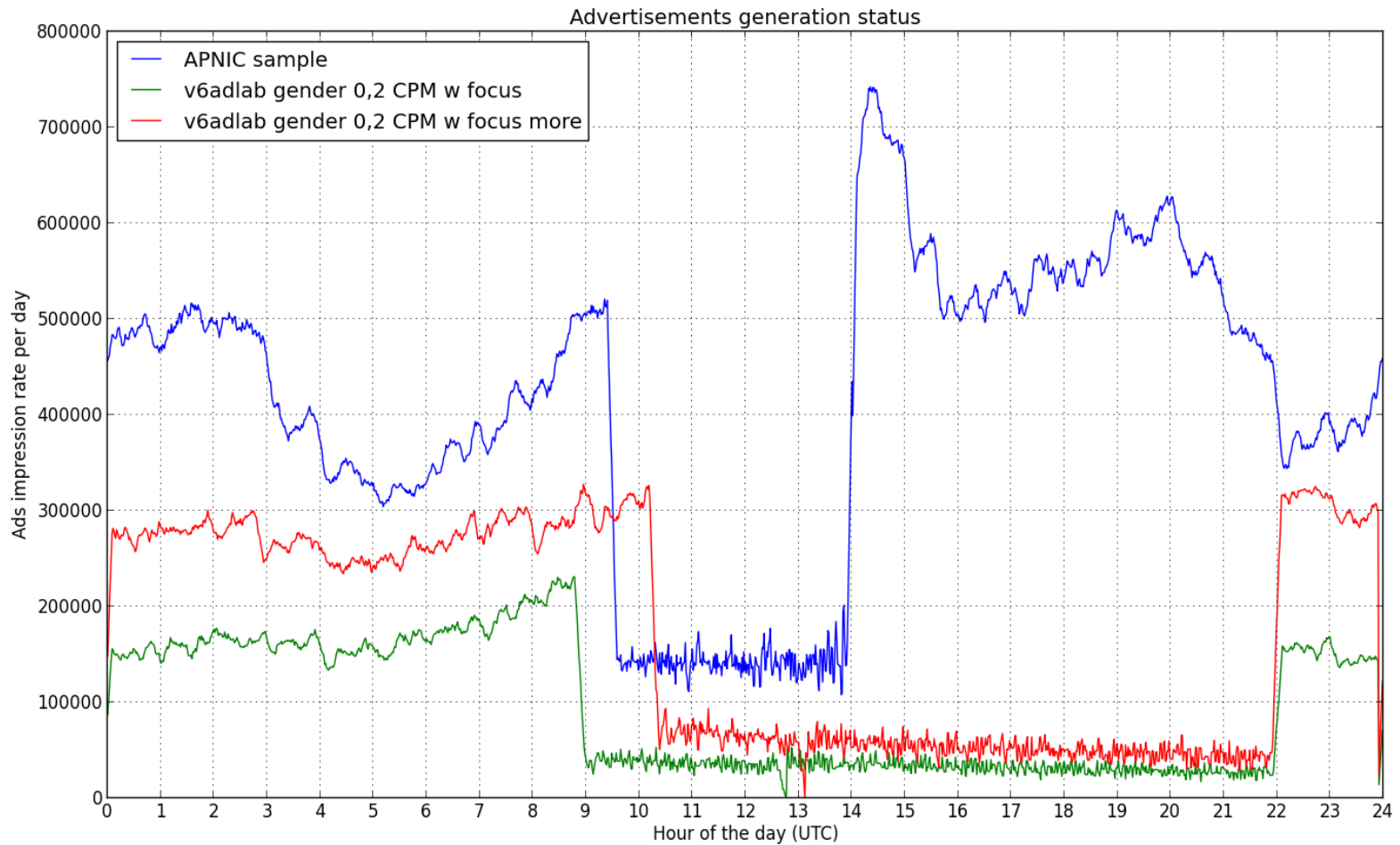


Space bias (replica)

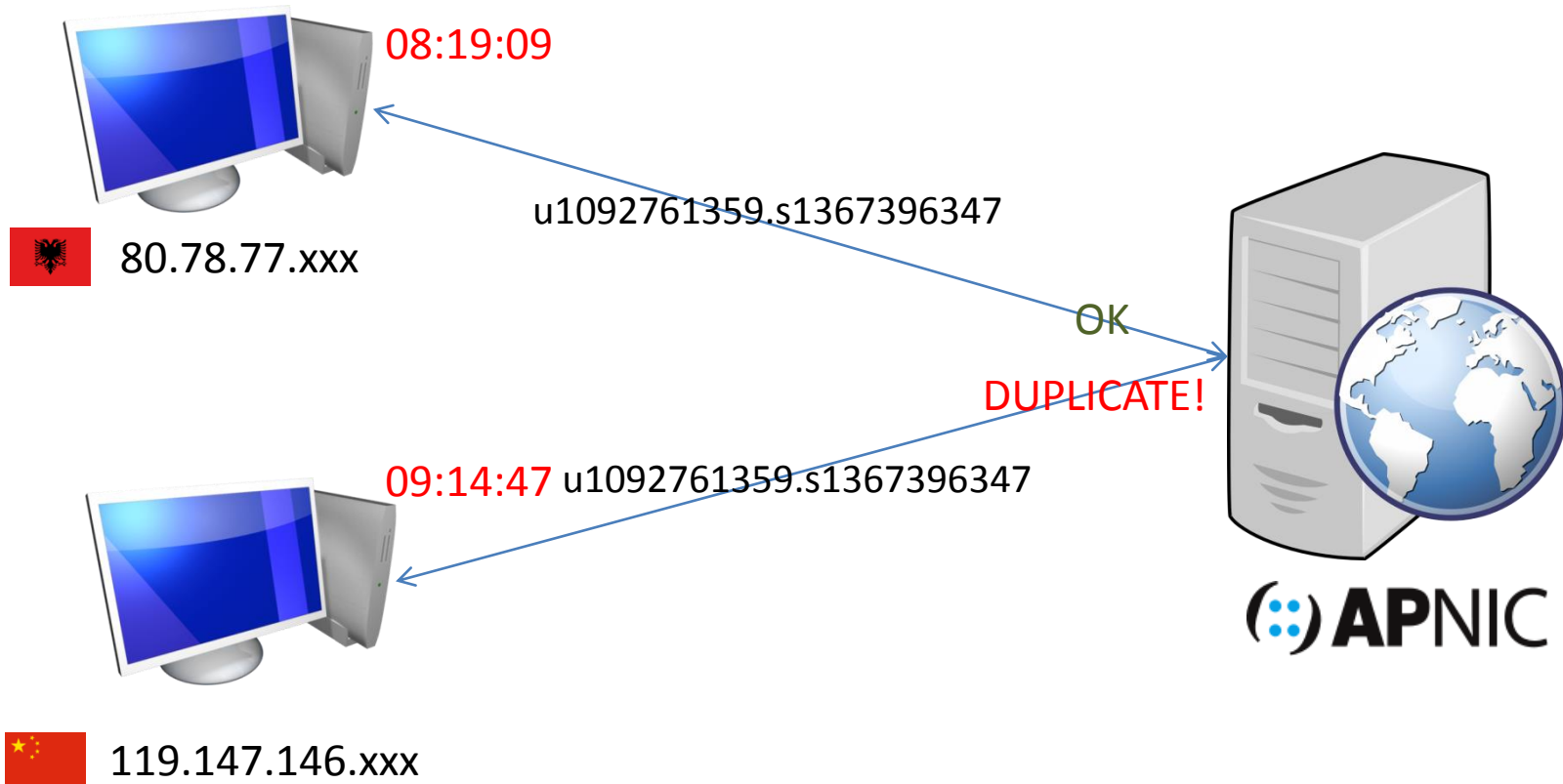
Time bias



Time bias



Unexpected results...



Fetching a AAAA-only DNS with IPv4, no problem!

dump-2013-07-13_04-54.pcap [Wireshark 1.8.2]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: http Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
4	0.126407	65.95.112	178.79.16	HTTP	371	GET /crossdomain.xml HTTP/1.1
6	0.127096	178.79.16	65.95.112	HTTP/XML	560	HTTP/1.1 200 OK
8	0.261134	65.95.112	178.79.16	HTTP	403	GET /measureipvid.php?s=Gcpm&ad=728&hash=3457062037 HTTP/1.1
9	0.261676	178.79.16	65.95.112	HTTP	489	HTTP/1.1 200 OK (text/plain)
20	1.073315	65.95.112	178.79.16	HTTP	409	GET /crossdomain.xml HTTP/1.1
22	1.073965	178.79.16	65.95.112	HTTP/XML	560	HTTP/1.1 200 OK
26	1.083546	65.95.112	178.79.16	HTTP	409	GET /crossdomain.xml HTTP/1.1
28	1.084097	178.79.16	65.95.112	HTTP/XML	560	HTTP/1.1 200 OK
33	1.207529	65.95.112	178.79.16	HTTP	450	GET /1x1.gif?t10000.u2629555544.s1373686280.i333.v10001.r4.td HTTP/1.1
34	1.207879	178.79.16	65.95.112	HTTP	391	HTTP/1.1 200 OK (GIF89a)
35	1.210533	65.95.112	178.79.16	HTTP	409	GET /crossdomain.xml HTTP/1.1
37	1.211145	178.79.16	65.95.112	HTTP/XML	560	HTTP/1.1 200 OK
39	1.222111	65.95.112	178.79.16	HTTP	450	GET /1x1.gif?t10000.u2629555544.s1373686280.i333.v10001.rd.td HTTP/1.1
40	1.222446	178.79.16	65.95.112	HTTP	391	HTTP/1.1 200 OK (GIF89a)
43	1.381162	65.95.112	178.79.16	HTTP	450	GET /1x1.gif?t10000.u2629555544.s1373686280.i333.v10001.r6.td HTTP/1.1
44	1.381481	178.79.16	65.95.112	HTTP	391	HTTP/1.1 200 OK (GIF89a)

▶ Frame 43: 450 bytes on wire (3600 bits), 450 bytes captured (3600 bits)

▶ Ethernet II, Src: 84:78:ac:5a:1a:41 (84:78:ac:5a:1a:41), Dst: f2:3c:91:70:07:8f (f2:3c:91:70:07:8f)

▶ **Internet Protocol Version 4**, Src: 65.95.112 (65.95.112.), Dst: 178.79.166.203 (178.79.166.203)

▶ Transmission Control Protocol, Src Port: 53387 (53387), Dst Port: http (80), Seq: 344, Ack: 495, Len: 384

▼ Hypertext Transfer Protocol

▶ GET /1x1.gif?t10000.u2629555544.s1373686280.i333.v10001.r6.td HTTP/1.1\r\n

Host: t10000.i333.v10001.9cbb-d558-51e0-ca08.r6.dyn.v6adlab.net\r\n

Connection: keep-alive\r\n

User-Agent: Mozilla/5.0 (X11; CrOS i686 3912.101.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36\r\n

Accept: */*\r\n

Accept-Encoding: gzip, deflate, sdch\r\n

Accept-Language: en-GB, en-US; q=0.8, en; q=0.6\r\n

\r\n

[Full request URI: <http://t10000.i333.v10001.9cbb-d558-51e0-ca08.r6.dyn.v6adlab.net/1x1.gif?t10000.u2629555544.s1373686280.i333.v10001.r6.td>]

0080 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 HTTP/1.1 .Host:
0090 74 31 30 30 30 30 2e 69 33 33 33 2e 76 31 30 30 t10000.i 333.v100
00a0 30 31 2e 39 63 62 62 2d 64 35 35 38 2d 35 31 65 01.9cbb- d558-51e
00b0 30 2d 63 61 30 38 2e 72 36 2e 64 79 6e 2e 76 36 0-ca08.r 6.dyn.v6

HTTP Host (http.host), 65 bytes Packets: 69 Displayed: 16 Marked: 0 Load time: 0:00.009 Profile: Default

Fetching a literal IPv6 with IPv4, no problem!

The image shows a Wireshark packet capture window titled "dump-2013-07-13_07-54.pcap [Wireshark 1.8.2]". The filter is set to "http". The packet list pane shows 54 packets, with packet 22 selected. The packet details pane shows the structure of the selected packet:

- Frame 22: 535 bytes on wire (4280 bits), 535 bytes captured (4280 bits)
- Ethernet II, Src: 84:78:ac:0d:8f:41 (84:78:ac:0d:8f:41), Dst: f2:3c:91:70:07:8f (f2:3c:91:70:07:8f)
- Internet Protocol Version 4 Src: 220.173.110. (220.173.110.) Dst: 178.79.166.203 (178.79.166.203)
- Transmission Control Protocol, Src Port: mpnjsomg (2686), Dst Port: http (80), Seq: 344, Ack: 495, Len: 481
- Hypertext Transfer Protocol
 - GET /1x1.gif?t10000.u3537280084.s1373695093.i333.v10001.v6lit HTTP/1.1\r\n
 - Accept: */*\r\n
 - Accept-Language: zh-CN\r\n
 - Referer: http://static.googleadserving.cn/pagead/imgad?id=CICAgIDQy5ajoQEQRrAIY-gEyCD0PcGFuZC1i\r\n
 - x-flash-version: 11,8,800,94\r\n
 - Accept-Encoding: gzip, deflate\r\n
 - User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET4.0C; .NET4.0E)\r\n
 - Host: [2a01:7e00:e000:3d:d2d6:9c54:51e0:ec75] \r\n
 - Connection: Keep-Alive\r\n
 - \r\n
 - [Full request URI: http://[2a01:7e00:e000:3d:d2d6:9c54:51e0:ec75]/1x1.gif?t10000.u3537280084.s1373695093.i333.v10001.v6lit]

The bottom pane shows the raw packet data in hexadecimal and ASCII:

```
0000 f2 3c 91 70 07 8f 84 78 ac 0d 8f 41 08 00 45 00 .<.p...x ...A..E.
0010 02 09 39 d2 40 00 35 06 66 4f dc ad 6e 05 b2 4f ..9.@.5. f0..n..0
0020 a6 cb 0a 7e 00 50 db ae 19 6d e8 48 50 41 50 18 ...~.P.. .m.HPAP.
0030 fe 11 0a f8 00 00 47 45 54 20 2f 31 78 31 2e 67 .....GE T /1x1.g
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```


Conclusion

- APNIC experiment can be reproduced
- Improvement on the geographical distribution, but not on the time distribution.
- Some hosts know how to use IPv4 to reach an IPv6 server, can you?

Thank you