Measuring the Effects of Happy Eyeballs

draft-bajpai-happy-01

Vaibhav Bajpai and Jürgen Schönwälder

{v.bajpai, j.schoenwaelder}@jacobs-university.de

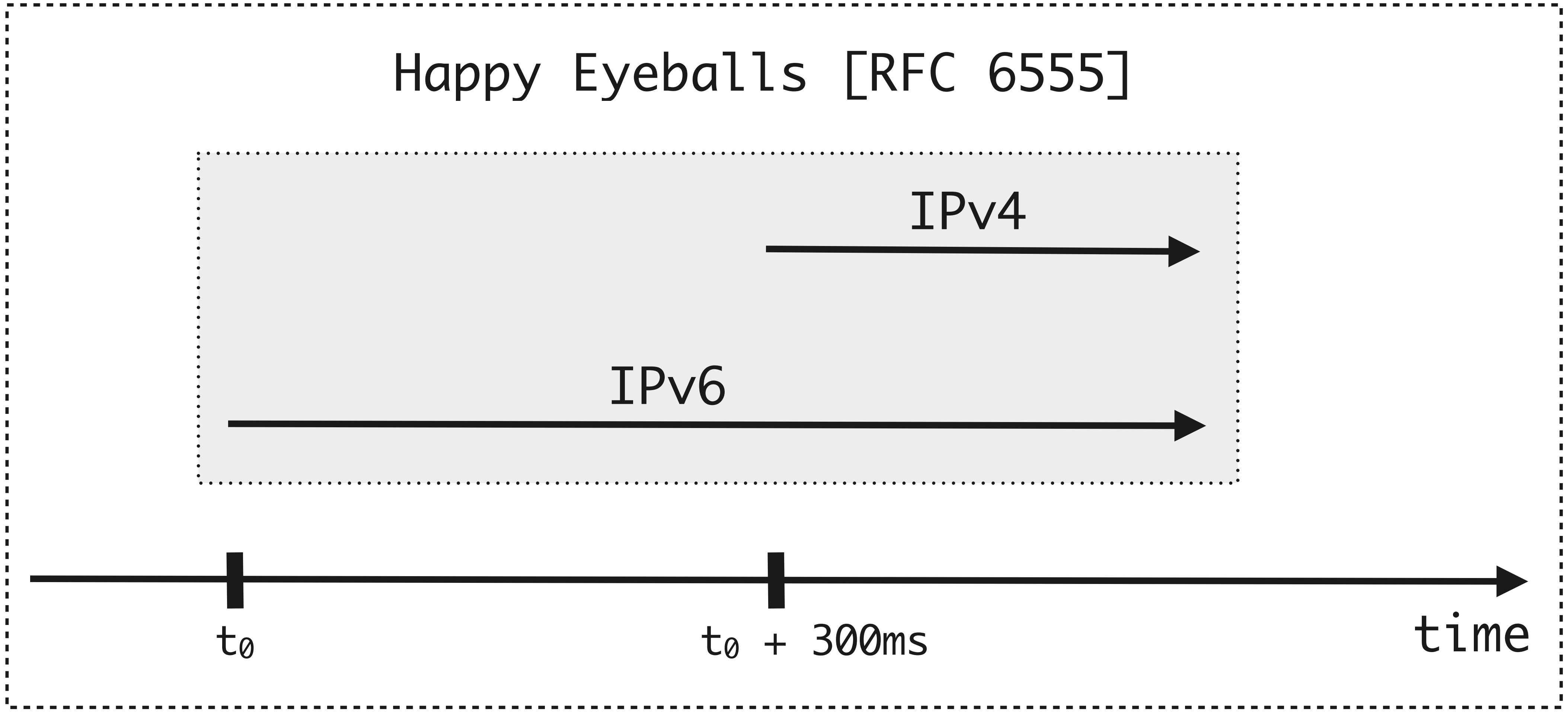
IETF 87, Berlin

Computer Networks and Distributed Systems
Jacobs University Bremen
Bremen, Germany

July 2013

Supported by:
Leone Project: http://leone-project.eu

Happy Eyeballs Algorithm [RFC 6555]



GOALS:

- Honor the destination address selection policy [RFC 6724].
- Quickly fallback to IPv4 when IPv6 connectivity is broken.
- Give a fair chance for IPv6 to succeed.

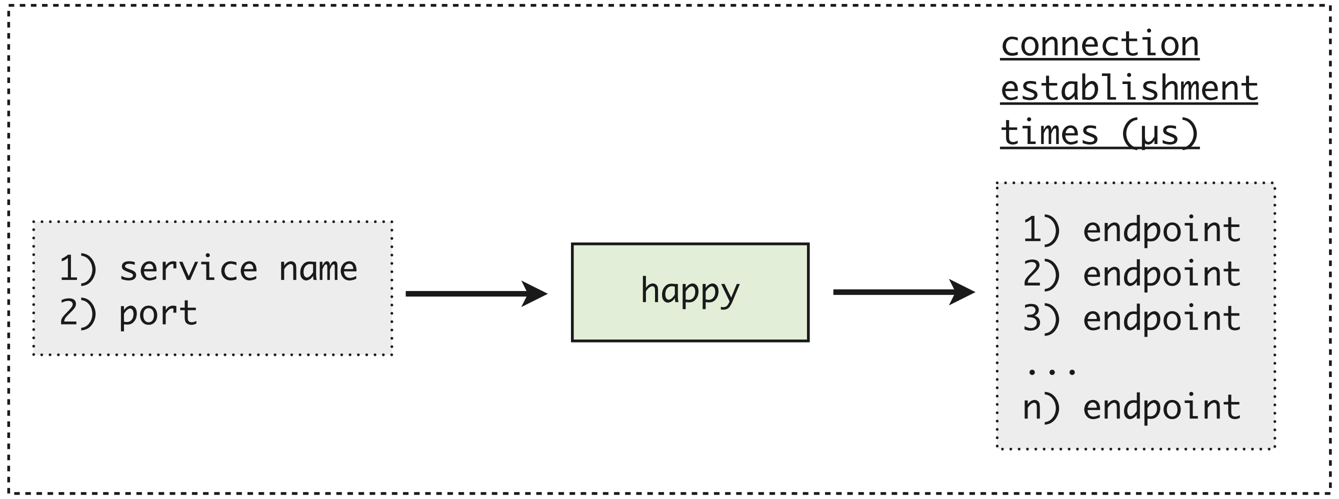
Research Question

- What is the amount of *imposition* a user experiences by turning on Happy Eyeballs?
  - [RFC 6555] will not be applied *only* in scenarios where IPv6 connectivity is broken.
  - How does it effect the experience of a dual-stacked host with *comparable* IPv6 connectivity?

- What is the *right* timer value?
  - [RFC 6555] recommends $150–250ms$.
  - Google Chrome uses $300ms$.
  - Firefox uses $250ms$.
  - Happy Eyeballs Erlang Implementation uses $100ms$: http://www.viagenie.ca/news/index.html#happy_eyeballs_erlang

Metrics and Implementation

- Uses `getaddrinfo(...)` to resolve service names.
- Uses non-blocking TCP `connect(...)` calls.
- DNS resolution time is not accounted.
- Capability to read multiple service names as arguments.
- Capability to read service names list from a file.
- File locking capability.
- Applies a delay between `connect(...)` to avoid SYN floods.
- Capability to produce both human-readable and CSV output.
- Cross-compiled for OpenWrt platform. Currently running from SamKnows probes.



http://happy.vaibhavbajpai.com

```
$ ./happy -q 1 -m www.google.com www.facebook.com
HAPPY.0;1360681039;OK;www.google.com;80;173.194.69.105;8626
HAPPY.0;1360681039;OK;www.google.com;80;2a00:1450:4008:c01::69;8884
```

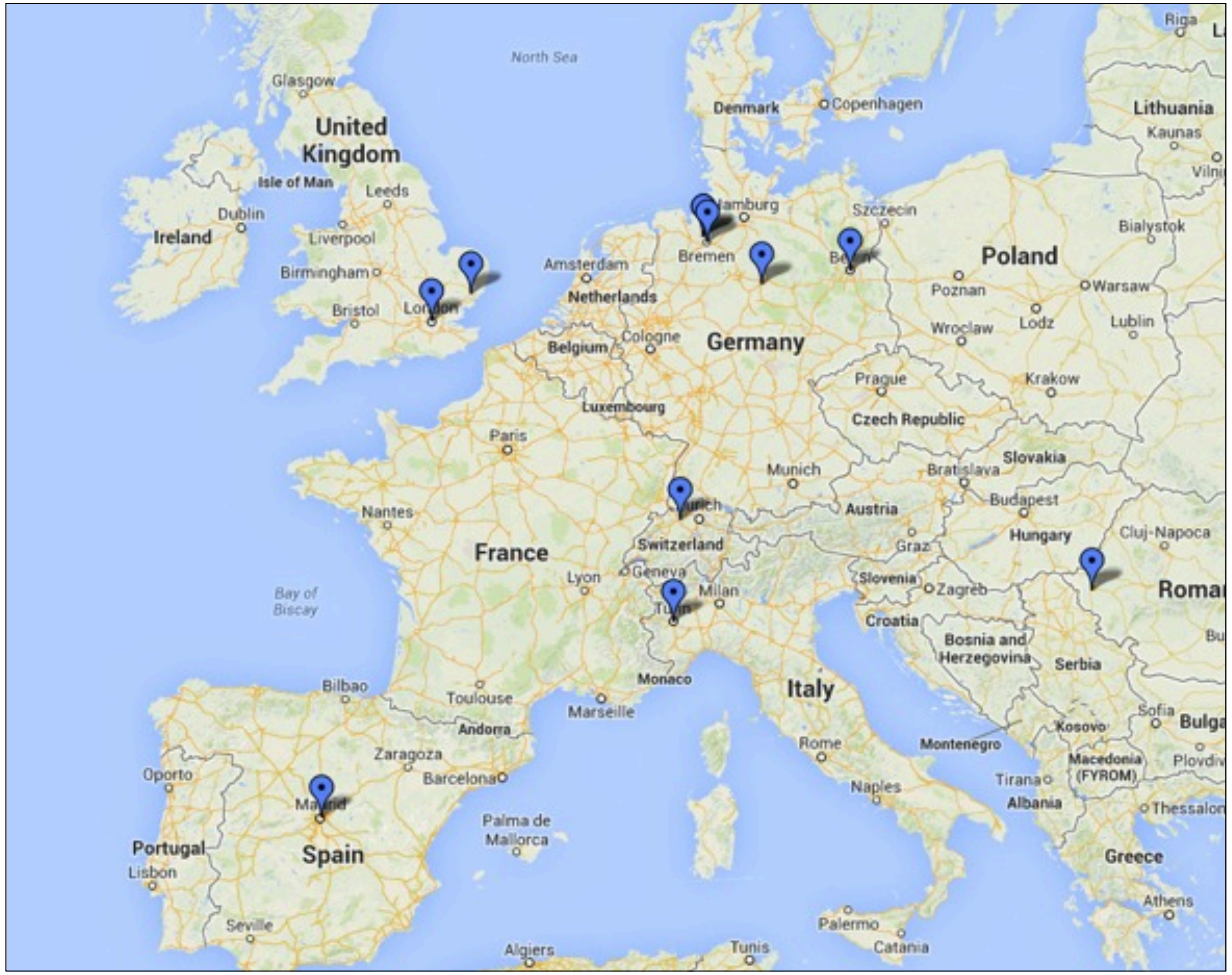
Measurement Trials

- How to *compile* a dual-stacked service names list?
  - Hurricane Electric (HE) maintains a top 100 dual-stacked service names list. http://bgp.he.net/ipv6-progress-report.cgi
    - HE uses top 1M service names list from Alexa Top Sites (ATS).
    - HE does *not* follow CNAMES.
  - Amazon has made the ATS top 1M service names list public. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip
    - Prepared a custom top 100 dual-stacked service names list.
    - Explicitly follow CNAMES.
    - Prepend a www to each service name and cross-check any AAAA response.

Measurement Trials

- From *where* to run the measurement test?

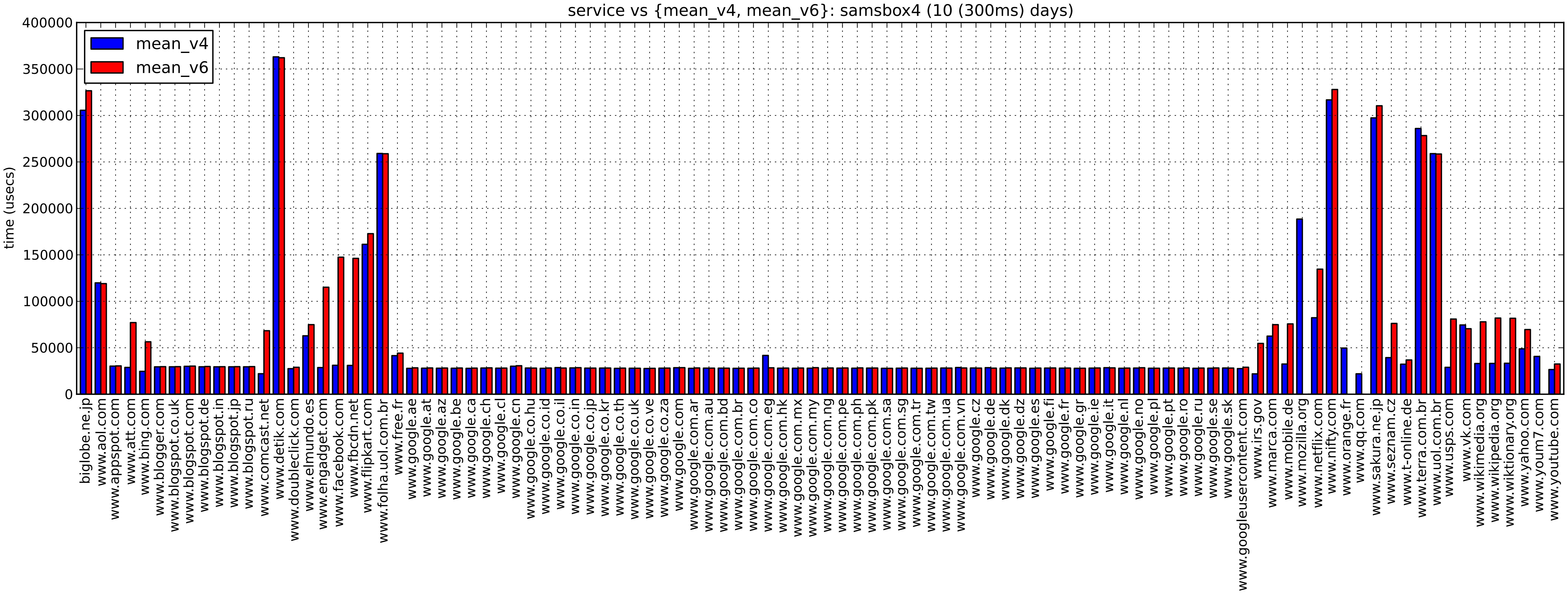
Provider (IPv4, IPv6)	Location
(Jacobs University Bremen, AS680), (-)	Bremen
(Kabel Deutschland, AS31334), (HE, AS6939)	Bremen
(Gaertner Datensystems GmbH, AS24956), (-)	Braunschweig
(Deutsche Telekom AG, AS3320), (-)	Bremen
(British Sky Broadcasting Limited, AS5607), (-)	London
(Telekom Italia, AS3269), (-)	Torino
(BT Spain, AS8903), (-)	Madrid
(ROEDUNET, AS2614), (-)	Timisoara
(Init Seven AG, AS13030), (-)	Olten
(BT-UK-AS, AS2856), (BT, AS5400)	Ipswich
(LambdaNet Communications, AS13237), (Teredo)	Berlin
(TU Braunschweig, AS24956), (-)	Braunschweig



(-) means the IPv6 provider and AS are same as that for IPv4.

Measuring Raw Performance

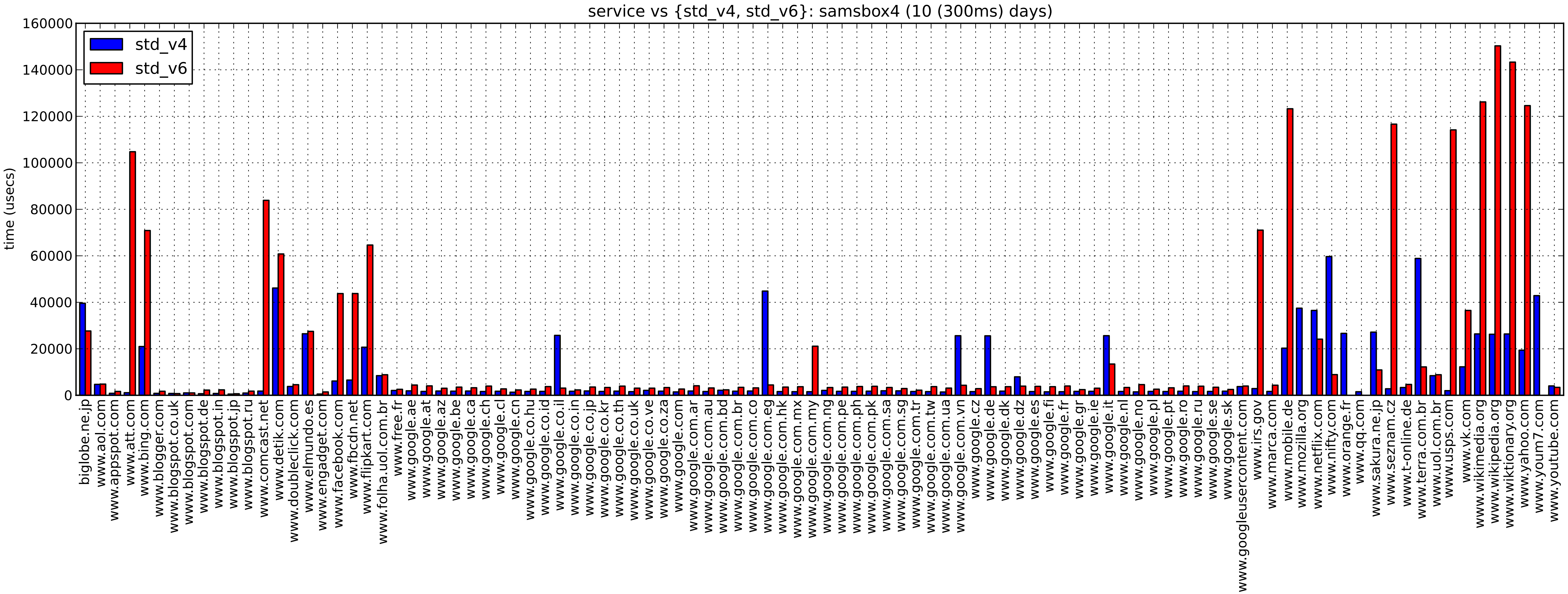
- How does the *performance (mean)* of IPv6 compare to that of IPv4?



Native IPv4 and IPv6 connectivity via DTAG - Deutsche Telekom AG [AS 3320]

Measuring Raw Performance

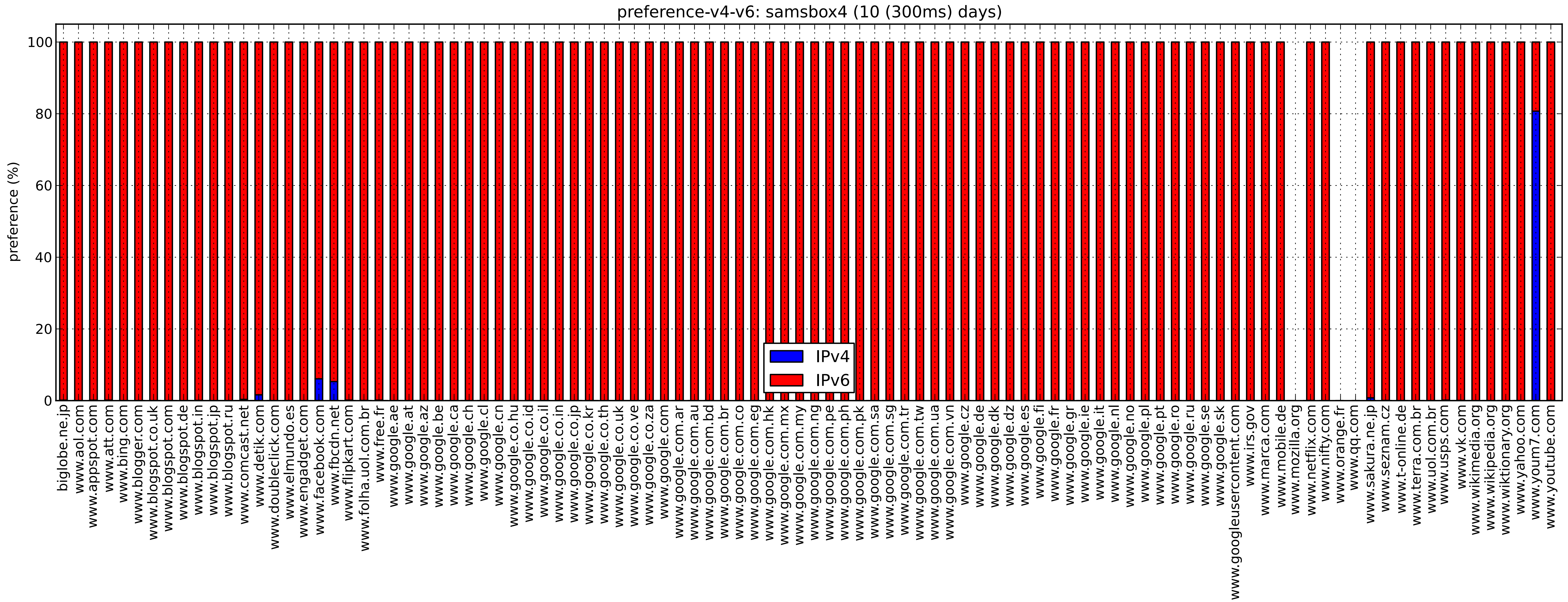
- How does the *performance (variation)* of IPv6 compare to that of IPv4?



Native IPv4 and IPv6 connectivity via DTAG - Deutsche Telekom AG [AS 3320]

Measuring Preference

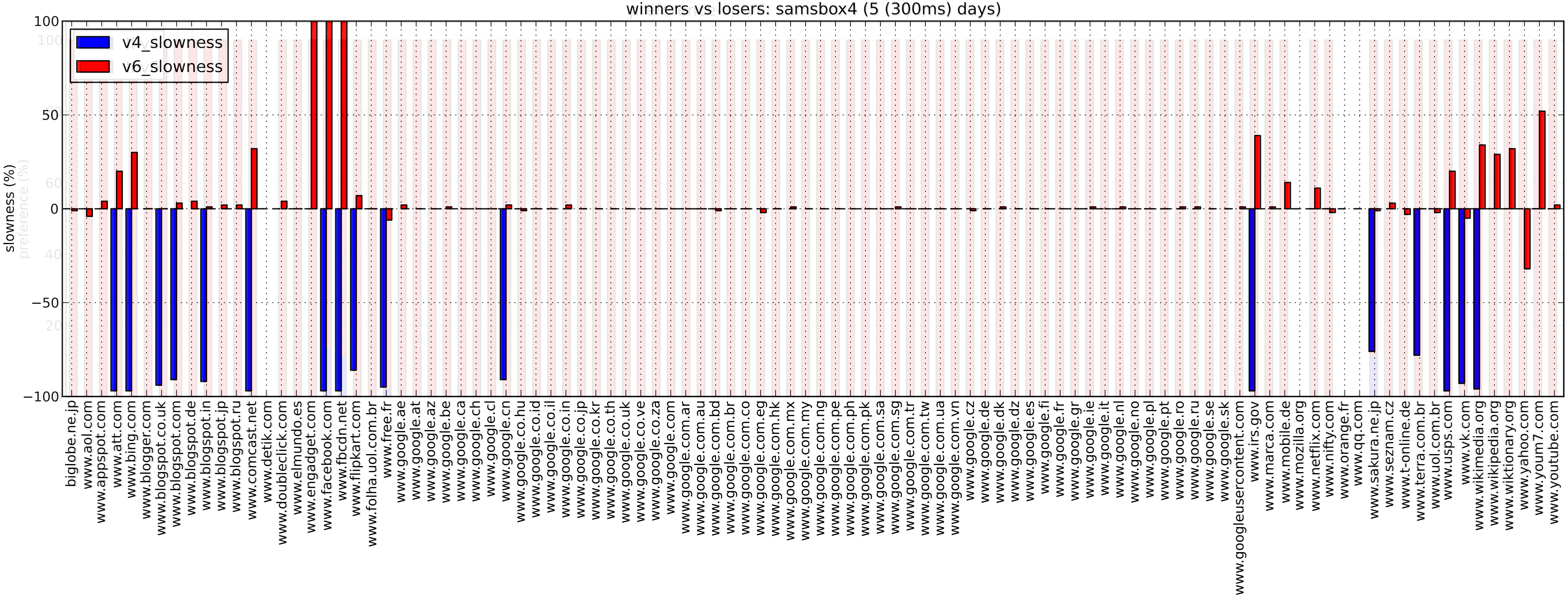
- To what extend is IPv6 *preferred* when connecting to a dual-stacked service?



Native IPv4 and IPv6 connectivity via DTAG - Deutsche Telekom AG [AS 3320]

Measuring Slowness

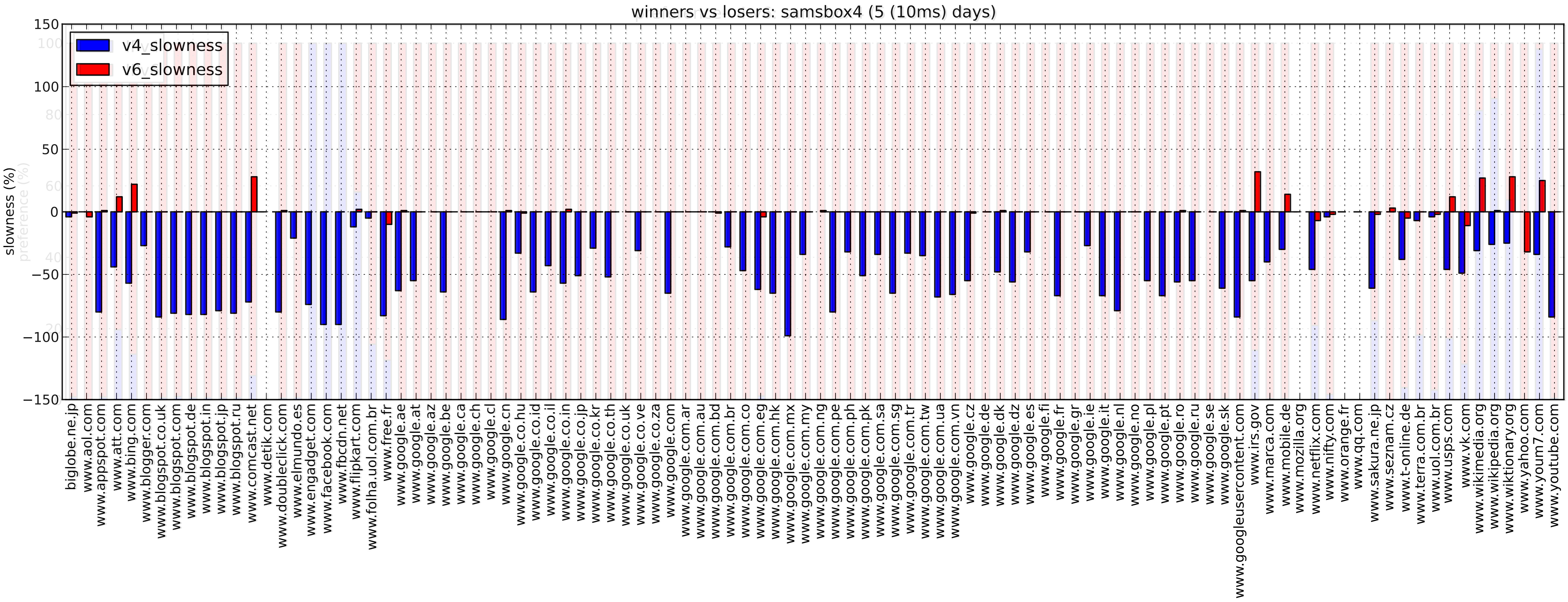
- How *slow* is a happy eyeballed winner to that of a loser?



Native IPv4 and IPv6 connectivity via DTAG - Deutsche Telekom AG [AS 3320]

Measuring Slowness

- What are the *repercussions* of reducing the IPv6 advantage from 300ms to 10ms?



Native IPv4 and IPv6 connectivity via DTAG - Deutsche Telekom AG [AS 3320]

Data Analysis Insights

- Higher connection times and variations over IPv6.
- A 300ms advantage leaves a MA 1% chance to prefer IPv4 (even though faster).
- A IPv6 happy eyeballed winner is rarely faster than the IPv4 route.
- A 10ms advantage helps remove outliers where IPv6 connectivity is bad.

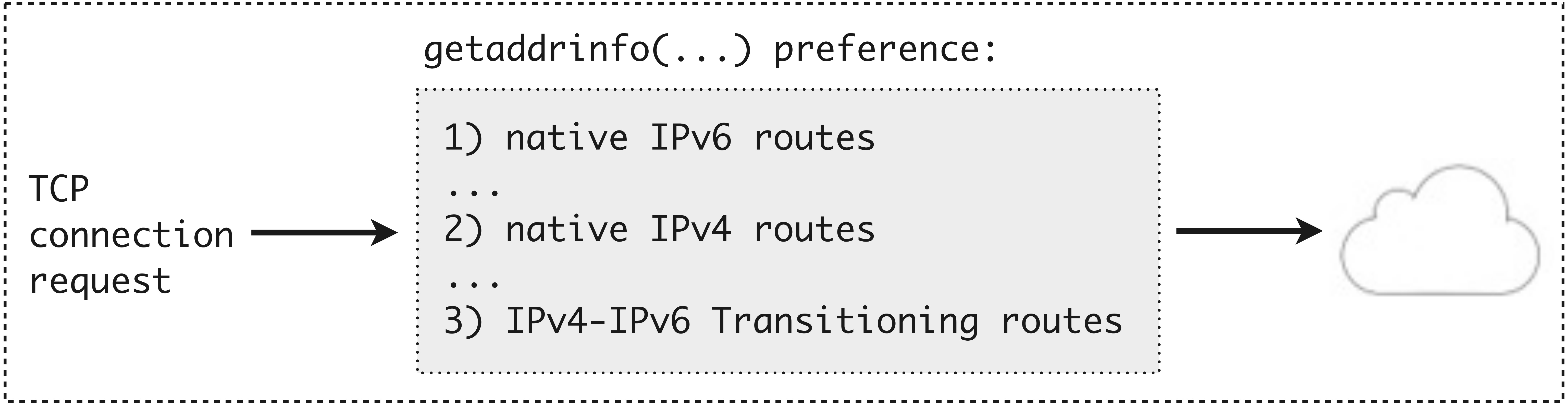
We would appreciate your help in our research activity:

- Send your shipment address to: v.bajpai@jacobs-university.de
- We ship you a SamKnows probe.

Appendix

getaddrinfo(...) behavior

- Returns a list of endpoints in an order that prioritizes IPv6-upgrade path.
- The order is dictated by [RFC 6724] and /etc/gai.conf
- If IPv6 connectivity is broken, an application is remains unresponsive for seconds.



IPv6 Upgrade Policy

- Why must IPv6 be given a *fair* chance to succeed?
    - Carrier Grade NAT (CG-NAT) creates a binding for each connection request.
        - reducing contention towards scarce IPv4 address space is desirable.
    - IPv4 traffic maybe billed by the Operation Support Systems (OSS).
        - moving traffic to IPv6 reduces network operation costs.
    - Middle-boxes maintain state for each connection request.
        - reducing load on peering links and load-balancers is desirable.

Related Work

- How is our measurement *different* from [RFC 6556]?

  - We do *not* account DNS in connection establishment time.
    - avoid input parameters that may *bias* the measurement (slow resolvers)

  - Our testbed configuration is *active* rather than passive.
    - measurement test *actively* measures time taken to establish the TCP connection.

  - Our testbed setup is designed for a *uncontrolled* environment.
    - does *not* require network path configuration changes.

Related Work

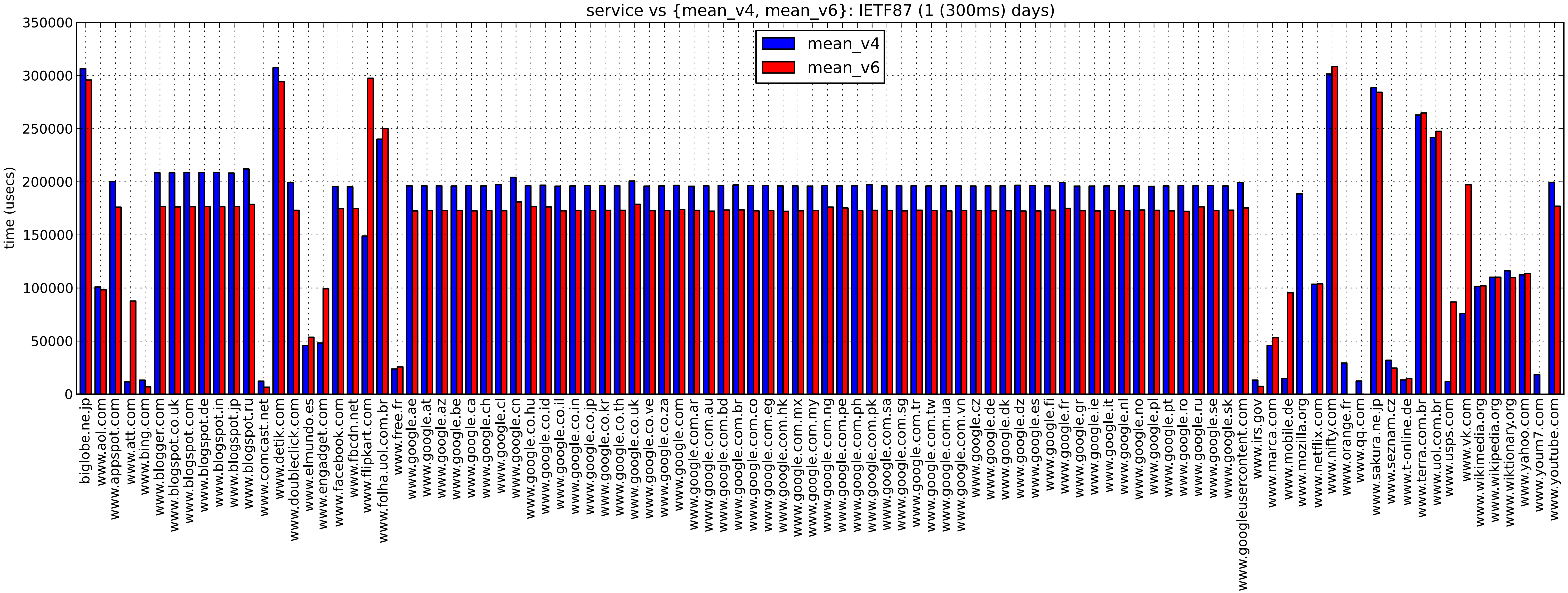
- How is our measurement *different* from [RFC 6948]?
  - Longer and *newer* measurement cycles.
    - [RFC 6948]: May 25, 2011 – July 11, 2011
    - We are running the measurement since Mar 10, 2013 – Present.
  - Measurement from a wider deployed vantage point
    - 3 MAs deployed somewhere in Finland, Sweden and Canada in [RFC 6948].
    - 14 MAs deployed across EU, more upcoming ...
  - We do *not* measure the amount of AAAA entries within 1M ATS.
    - [RFC 6948] noticed around 300 (within top 10K ATS) services were dual stacked.
    - [RFC 6948] noticed around 30 (within top 100 ATS) services were dual stacked.
    - We take top 1M ATS and filter the top 100 dual-stacked services.

Related Work

- How are our measurement results different from [RFC 6948]?
  - We noticed significantly *higher* TCP connection setup delay differences.
    - Generally slower over IPv6.
    - Multiple services were twice as slow over IPv6 when compared to IPv4.
  - We noticed significantly *lower* TCP connection setup failure rates.
    - We witnessed 1% of service failure rates, as opposed to 20% witnessed in [RFC 6948].
  - We perform a *deeper* TCP connection setup delay study.
    - Take happy eyeballs effects into account.
    - Measure the routing path differences over IPv4 and IPv6.

Measuring Raw Performance

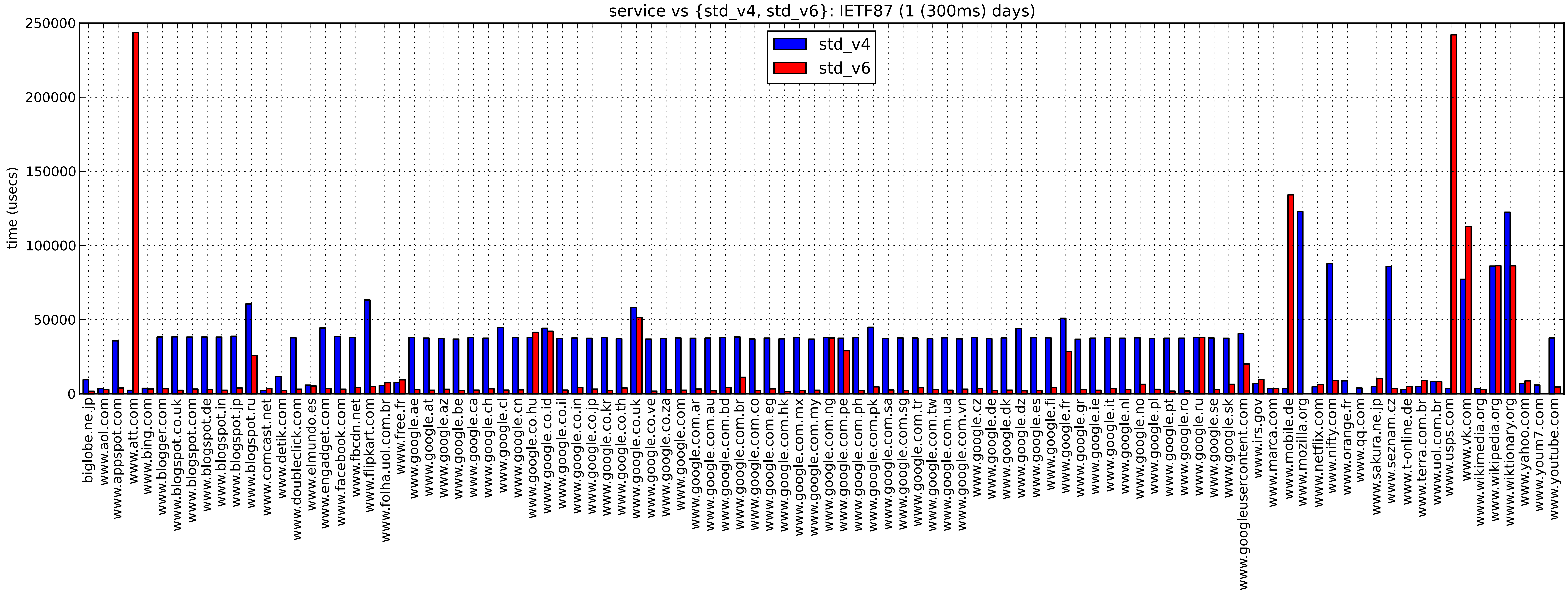
- How does the *performance (mean)* of IPv6 compare to that of IPv4?



Native IPv4 and IPv6 connectivity via IETF-Meeting - Internet Society [AS 56554]

Measuring Raw Performance

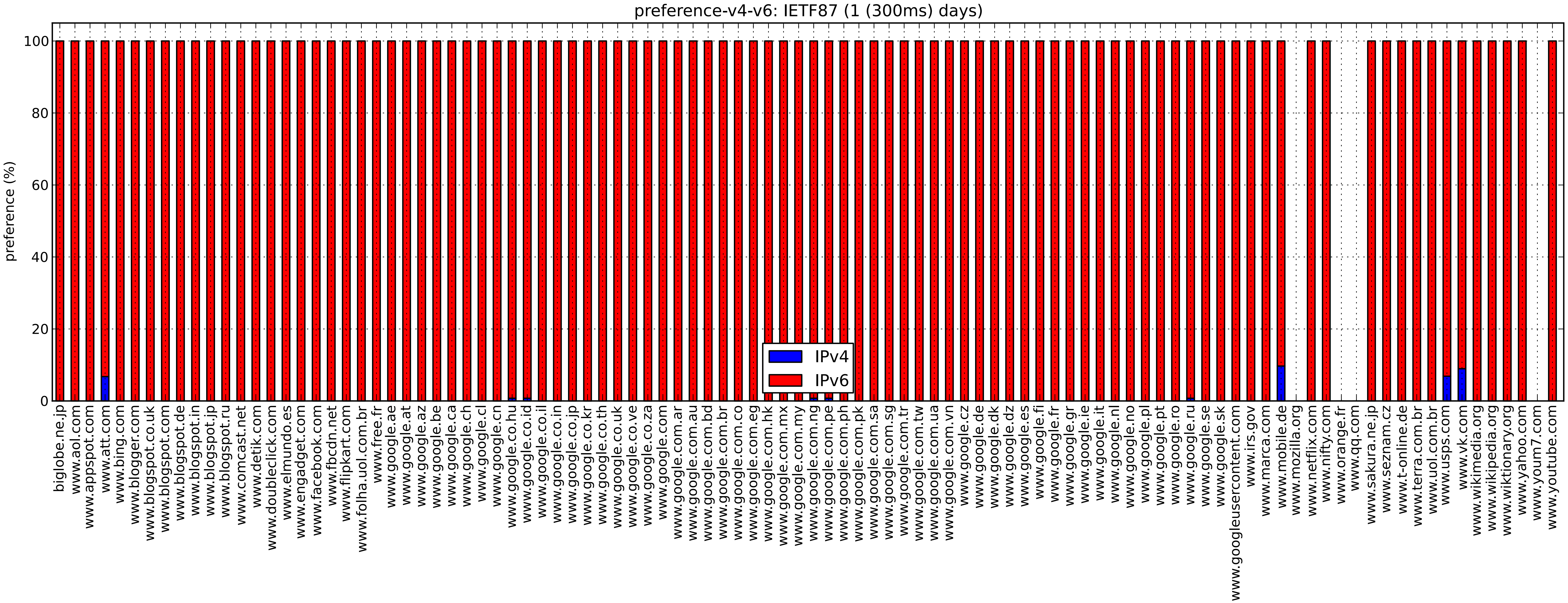
- How does the *performance (variation)* of IPv6 compare to that of IPv4?



Native IPv4 and IPv6 connectivity via IETF-Meeting - Internet Society [AS 56554]

Measuring Preference

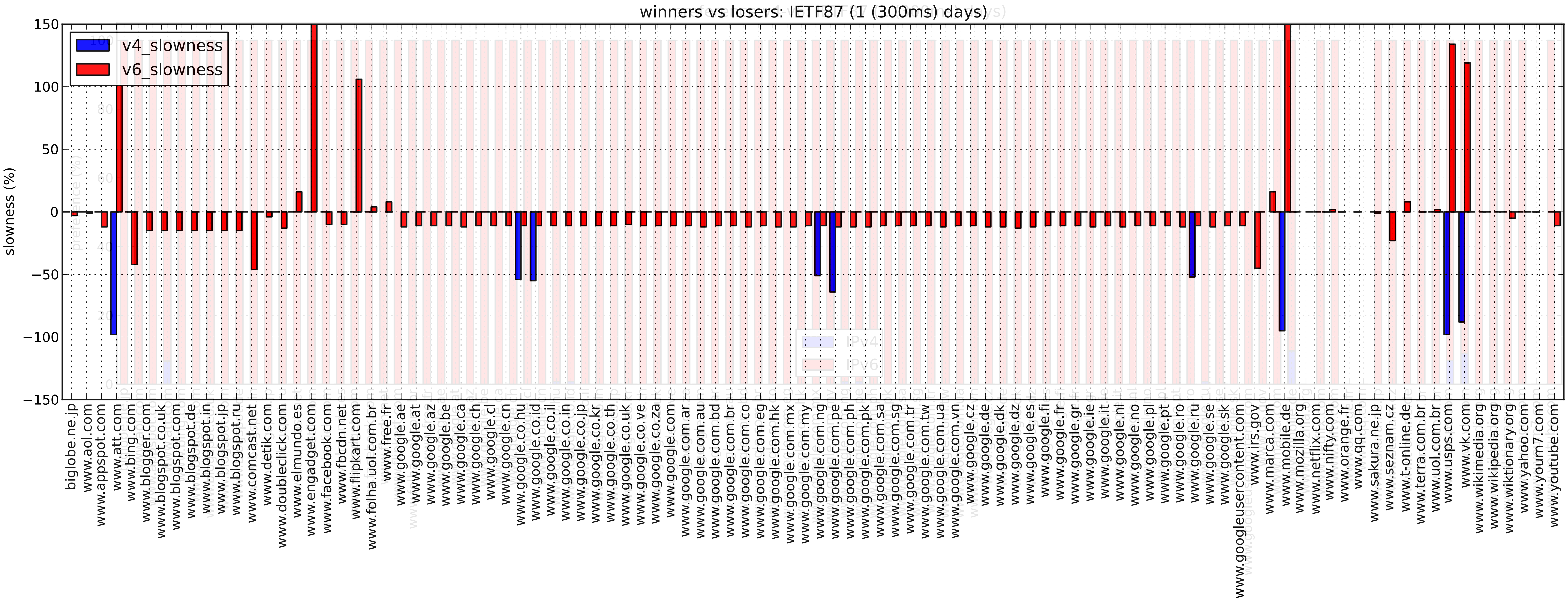
- To what extend is IPv6 *preferred* when connecting to a dual-stacked service?



Native IPv4 and IPv6 connectivity via IETF-Meeting - Internet Society [AS 56554]

Measuring Slowness

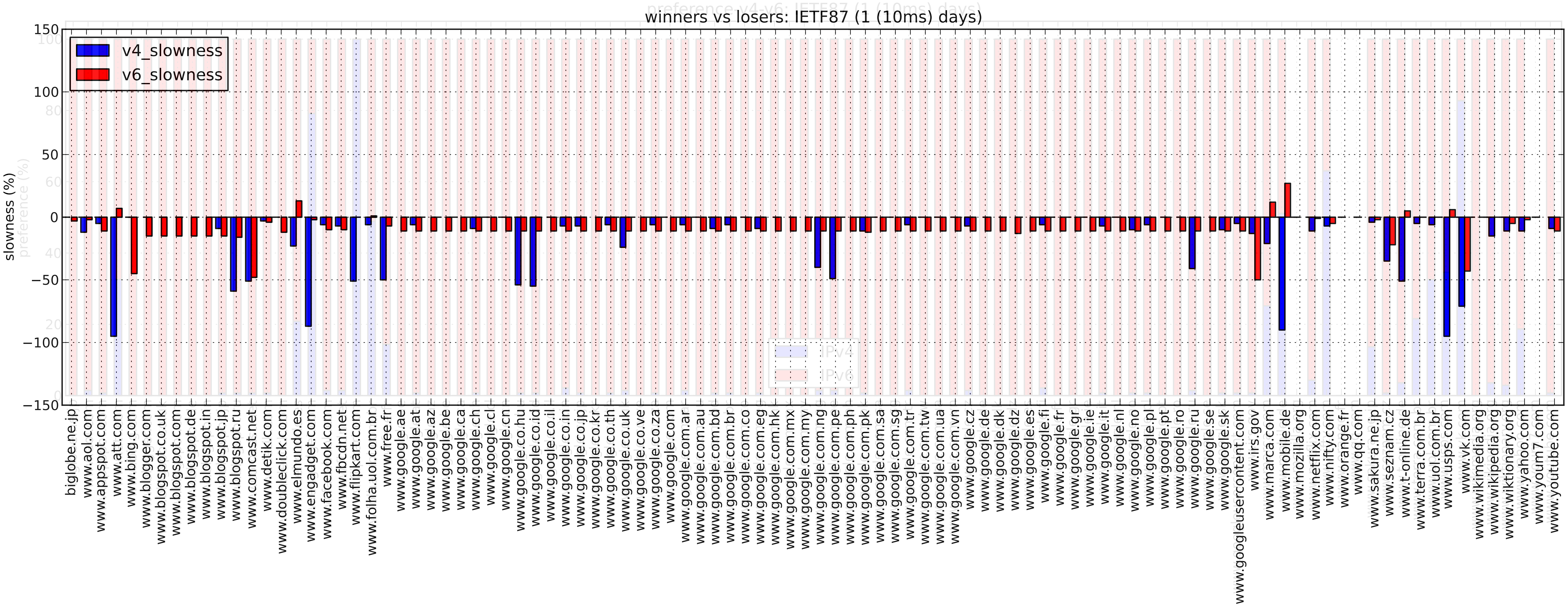
- How *slow* is a happy eyeballed winner to that of a loser?



Native IPv4 and IPv6 connectivity via IETF-Meeting - Internet Society [AS 56554]

Measuring Slowness

- What are the *repercussions* of reducing the IPv6 advantage from 300ms to 10ms?



Native IPv4 and IPv6 connectivity via IETF-Meeting - Internet Society [AS 56554]