iOS OSx and IPv6

Exploring the Impact of IPv6 Preference

Chris Baker

November 19th RIPE 71


I WANT YOU
TO USE IPv6
– VINT CERF
2015–09–16
ONE WAY

BACKGROUND

- On July 6, 2015 Apple announced on an IETF mailing list that they were going to explore the impact on their platform of implementing an IPv6 preference.
- David Schinazi described how they included an improved version of their implementation of "Happy Eyeballs” in iOS 9 and OSx El Capitan
- **“Please test this out if you have the means to, we'd love to see test results and receive feedback!”**
  - What follows is an attempt to explore the potential impact of this change in behavior

https://www.ietf.org/mail-archive/web/v6ops/current/msg22455.html

The Preference Implementation

Step 1: The device will query the stub resolver for A (IPv4) and AAAA (IPv6) records.

- If the records are not in cache, then requests will be sent to the recursive resolver in succession starting with the AAAA record.

Step 2: If the first reply the device receives is a AAAA, then it will send out a v6 SYN immediately. If the first reply the device receives is an A record a 25ms timer is started, the device is essentially waiting for the AAAA.

- If the AAAA response is received before the 25ms timer finishes, the device advances to address selection (for details on address selection see IETF mailing list link).

What can be tested in the wild?

- **IPv4 vs. IPv6 Network Topology:**

1. Will a DNS query over IPv6 to an anycasted address end up at a different location than one over IPv4 to an anycasted address?
    - What is the impact on latency?
2. If they do end up in different locations, can we learn anything from looking at the paths they take?

- **Recursive Resolvers and Negative Caching**

1. Is the 25ms “V6” tax real for non-dual stack applications?
2. Do recursive resolvers properly cache the NO DATA response they receive when a AAAA request is made for a resource for which one doesn’t exist?

To
RIPE Atlas!

IPv4 vs. IPv6 Network Topology

- **Dyn operates a global anycast network**
  - We utilize large global ISPs to maintain stable routes and announcements
  - No IX Peering of nameserver prefixes
- **For the first round of testing:**
  - Select a population of probes from a similar geographic area
  - Create one IPv4 and one IPv6 DNS measurement
    - The measurement consists of a CHAOS query to a specific target IP which is being announced over the same provider from a number of locations
    - The queries are being sent directly our nameservers to ensure it’s the routing topology being tested
  - Review results and repeat for a variety of geographic areas

Note:

These observations are specific to Dyn's network

Anomalies

- **Oceania**
  - Some IPv6 queries were being routed to the West Coast of the US
    - IPv4 stayed in the region where as IPv6 traffic was routed to West Coast of the US
- **South America**
  - IPv6 queries were routed to the north-eastern coast of the US (New Jersey / Virginia)
    - IPv4 queries were routed to on continent datacenters (Brazil) and Miami
- **Africa**
  - For IPv6 tunnels were terminating in Europe and v6 traffic using transit went to Europe
  - For IPv4 mix of traffic landing in Europe and Asia

A Few Examples: Oceania

Of 28 probes in Australia / New Zealand 8 end up at different anycast endpoints (2 probes have equally worrisome destinations)

Probe 22666 and 17567

- Both V4 and V6 queries were answered by servers on the West Coast of the US
- V4: ~147.7 ms V6: ~ 148.2 ms

Probes 22734, 1022, 11753, 20315, 10933

- V4 queries were answered by in Sydney V6 were answered by Palo Alto
- The latency difference in these cases is > 120 ms
  - Example V4: ~ 33.4 ms V6: ~ 160 ms V4: ~12.5 ms V6: ~ 170 ms

A Few Examples: Middle East - Asia

Probe 17450 – Malaysia

- V4 were answered by London V6 were answered by Singapore
- V4: ~252.5 ms V6: ~ 185 ms

Probes 19127 - Philippines

- V4 were answered by Palo Alto V6 were answered by Singapore
- V4: ~ 160 ms V6: ~ 52 ms

Probes 6128 - Pakistan

- V4 were answered by Frankfurt V6 were answered by Palo Alto
- V4: ~ 33.4 ms V6: ~ 160 ms

Probe 17312 – Japan

- V4 were answered by Toyko V6 were answered by Hong Kong
- V4: ~24 ms V6: ~ 85 ms

Probe 22534 – China

- V4 were answered by Toyko V6 were answered by Los Angeles
- V4: ~161 ms V6: ~ 170 ms

Probes 11287 - Japan

- V4 were answered by Tokyo V6 were answered by Singapore
- V4: ~ 8.2 ms V6: ~ 290 ms

A Few Examples: South America

Probe 23500- Uruguay

- V4 were answered by Miami V6 were answered by Los Angeles
- V4: ~141 ms V6: ~ 224 ms

Probes 4939- Argentina

- V4 were answered by Miami V6 were answered by Ashburn
- V4: ~ 210 ms V6: ~ 270 ms

Probes 16721- Brazil

- V4 were answered by Miami V6 were answered by Ashburn
- V4: ~ 175 ms V6: ~ 202 ms

IPv4 vs. IPv6 Network Topology

1. Will a DNS query over IPv6 to an anycasted address end up at a different location than one over IPv4 to an anycasted address?
    - In a non-trivial number of samples the answer is yes

- What is the impact on latency?
    - This depends on how different the paths are. In a number of cases the IPv6 latency was large enough that the “V6 Tax” was real

IPv4 vs. IPv6 Network Topology

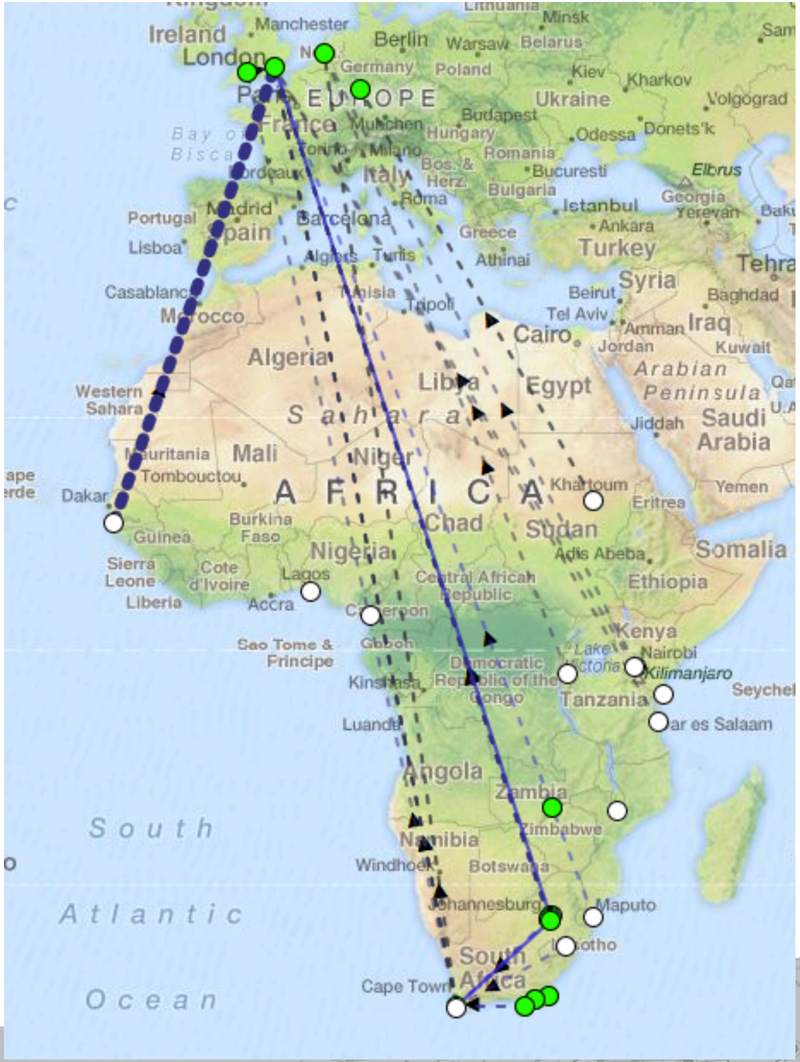
2. If they do end up in different locations, can we learn anything from looking at the paths they take?

Data collection for observing path difference

- **For the probes which reach different endpoints**
  - Run a traceroute
- **For the first round of testing:**
  - Enter the probe ids which had the divergent IPv4 and IPv6 paths
  - Create one IPv4 and one IPv6 traceroute measurement
  - Look at where the paths diverge

IPv4 vs IPv6 in Africa

The woes of GeoIP location aside, would you assume that these are IPv4 or IPv6 traceroutes to anycasted addresses?




Ireland
Manchester
London
Berlin
Warsaw
Belarus
Minsk
Samara
Germany
Poland
Kiev
Kharkov
Paris
EUROPE
Ukraine
Volgograd
Kazakhstan
France
München
Budapest
Hungary
Odessa
Donets'k
Bay of Biscay
Milano
Torino
Romania
Bordeaux
Italy
Bucuresti
Elbrus
Aral Sea
Uzbekistan
Almaty
Urumqi
Mongolia
Roma
Bulgaria
Istanbul
Georgia
Yerevan
Baku
Kyrgyzstan
Gobi
Madrid
Barcelona
Ankara
Turkmenistan
Toshkent
Portugal
Spain
Greece
Tajikistan
Lisboa
Algiers
Tunis
Athinai
Turkey
Tehran
Mashhad
Lanzhou
Kabul
China
Casablanca
Tunisia
Tripoli
Beirut
Syria
Baghdad
Iran
Afghanistan
Morocco
Tel Aviv
Amman
Iraq
Lahore
Cairo
Jordan
Kuwait
Chongqing
Delhi
Everest
Algeria
Arabian Peninsula
Pakistan
Kanpur
Patna
Western Sahara
Libya
Egypt
Qatar
Karachi
Jaipur
U.A.E.
Masqat
Ahmadabad
Kolkata
Dhaka
Sahara
Saudi Arabia
Jiddah
Oman
India
Myanmar
Hanoi
Mauritania
Mali
Niger
Mumbai
Hyderabad
Laos
Tombouctou
Arabian Sea
Bay of Bengal
Vientiane
Dakar
AFRICA
Khartoum
Eritrea
Yemen
Thailand
Burkina Faso
Chad
Sudan
Bangalore
Chennai
Bangkok
Cambodia
Guinea
Nigeria
Adis Abeba
Somalia
Phnom Penh
Sierra Leone
Cote d'Ivoire
Lagos
Central African Republic
Ethiopia
Colombo
Sri Lanka
Liberia
Accra
Cameroon
Maldives
Kuala Lumpur
Kenya
Singapore
Sao Tome & Principe
Gabon
Nairobi
Malaysia
Democratic Republic of the Congo
Kilimanjaro
Seychelles
Kinshasa
Tanzania
Jakarta
Dar es Salaam
Bandung
Luanda
Indian
Angola
Zambia
South
Zimbabwe
Mauritius
Ocean
Namibia
Windhoek
Botswana
Atlantic
Johannesburg
Maputo
South Africa
Cape Town
Ocean

Looking at the Middle East

IPv4 - Paths

23966 -> 17557 -> 6762 -> 2914 -> 33517

RTT 167 – Lands in Frankfurt

IPv6 - Paths

23966 -> 38193 -> 3356 -> 6453 -> 33517

RTT 150 – Lands in Amsterdam

6	2404:d400:0:24::1	38193	(none)
7	2001:1900:5:2:2::25cd	3356	xe-8-1-3.edge4.Frankfurt1.Level3.net
8	2001:1900:5:1::212	3356	vl-4060.edge4.Dusseldorf1.Level3.net
9	2001:1900:5:1::111	3356	vl-4080.edge3.Dusseldorf1.Level3.net
10	2001:1900:5:1::20d	3356	vl-4040.edge3.Amsterdam1.Level3.net
11	2001:1900:102:1::d	3356	vl-51.ear2.Amsterdam1.Level3.net
12	2001:1900:4:3::266	3356	Tata-level3-40G.Amsterdam.Level3.net
13	2a01:3e0:ff40:200::21	6453	if-ae11.2.tcore2.AV2-Amsterdam.ipv6.as6453.net
14	2a01:3e0:ff40:200::76	6453	(none)

Hurricane Electric Tunnels

When looking at the traceroute data a number of the divergent samples contained a Hurricane Electric Tunnel

- Tunnels signified by ###.ipv6.he.net
- Example: tunnel.tserv12.mia1.ipv6.he.net

Whether these tunnels are knowingly configured to cause traffic to take a suboptimal path or if its done for other reasons, they impact where traffic is being routed

hop	IP	ASN	hostname	RTTs
1	2001:470:5:471::1	6939	mikrotik.globalitss.com	0.5 0.5 0.5
2	2001:470:4:471::1	6939	Nickmman-1.tunnel.tserv12.**mia1.ipv6.he.net**	176.1 176.8 176.8
3	2001:470:0:90::1	6939	ge5-4.core1.**ash1**.he.net	171.4 177.1 174.8
4	2001:470:0:191::2	6939	(none)	171.2 171.2 171.2
5	2001:5a0:600:400::9d	6453	if-ae16.2.tcore1.AEQ-**Ashburn**.ipv6.as6453.net	192.8 193.9 190.5

Path Variation

It’s safe to say that the IPv6 default may change the results devices have been receiving from the DNS

IPv4 and IPv6 DNS queries issued from the same device can end up at different resolvers

- If people are using anycast to shape response semantics, IPv4 and IPv6 may return different results

It’s worth spending some additional time to understand and measure the differences and how they might impact the customer experience

Recursive Resolvers and Negative Caching

1. Is the 25ms “V6” tax real for non-dual stack applications?
2. Do recursive resolvers properly cache the NO DATA response they receive when a AAAA request is made for a resource for which one doesn’t exist?

Looking At Resolvers

- This set of tests is based on a pessimistic view and presupposes that non RFC conforming DNS resolution implementations may cause issues
- Will the 25ms delay introduced to wait for the AAAA response impact performance, creating a V4 only tax?
  - If a recursive doesn't properly cache NO DATA responses and additional recursion is required the 25ms tax sounds realistic

Looking At Resolvers

Using Atlas probes DNS measurement "Use probe local resolver setting" to explore the NO DATA caching.

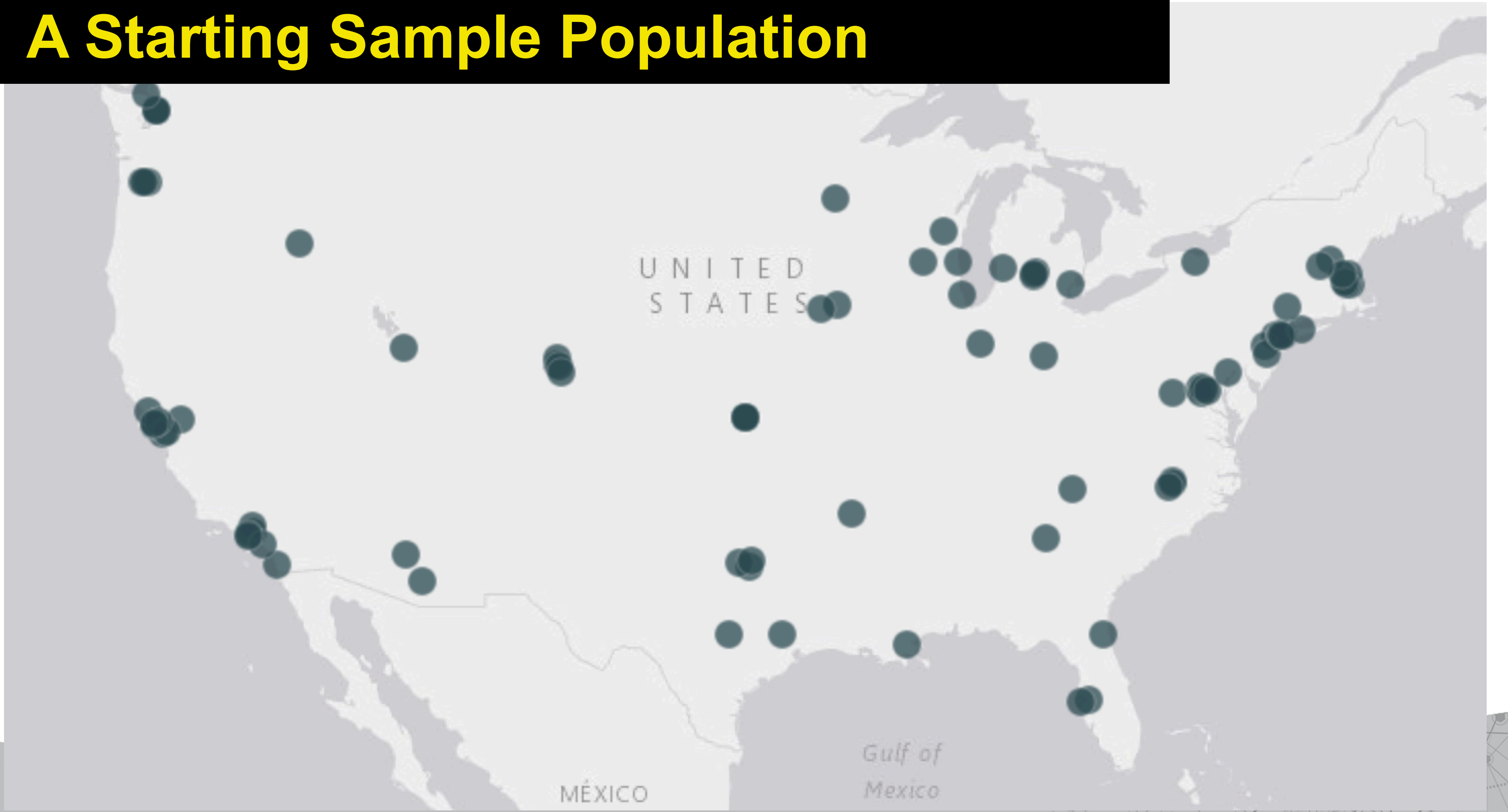
- Create testing domains with TTLs configured significantly long enough compared to the sampling rate to detect negative caching.
  - Example: If testing interval is 1800 then the TTL on the record needs to be smaller help with the observation of cache hit / miss patterns
- Test Cases
  - A domain with A (V4) and AAAA (V6) resources configured
  - A domain with only a V4 (A) resource record
  - A domain with only a V6 (AAAA) resource record

Dual Stack Application

Create two different user defined measurements one performing A queries the other AAAA queries

- If the AAAA resolves in < 25 ms (or faster than than the A) an Apple device it would issue the V6 SYN
- If the AAAA takes significantly longer to resolve than the A the difference in latency would be a measure of the V6 tax
- The queries aren’t being issues at the same time which introduces area for error but the test provides some directional cover

A Starting Sample Population

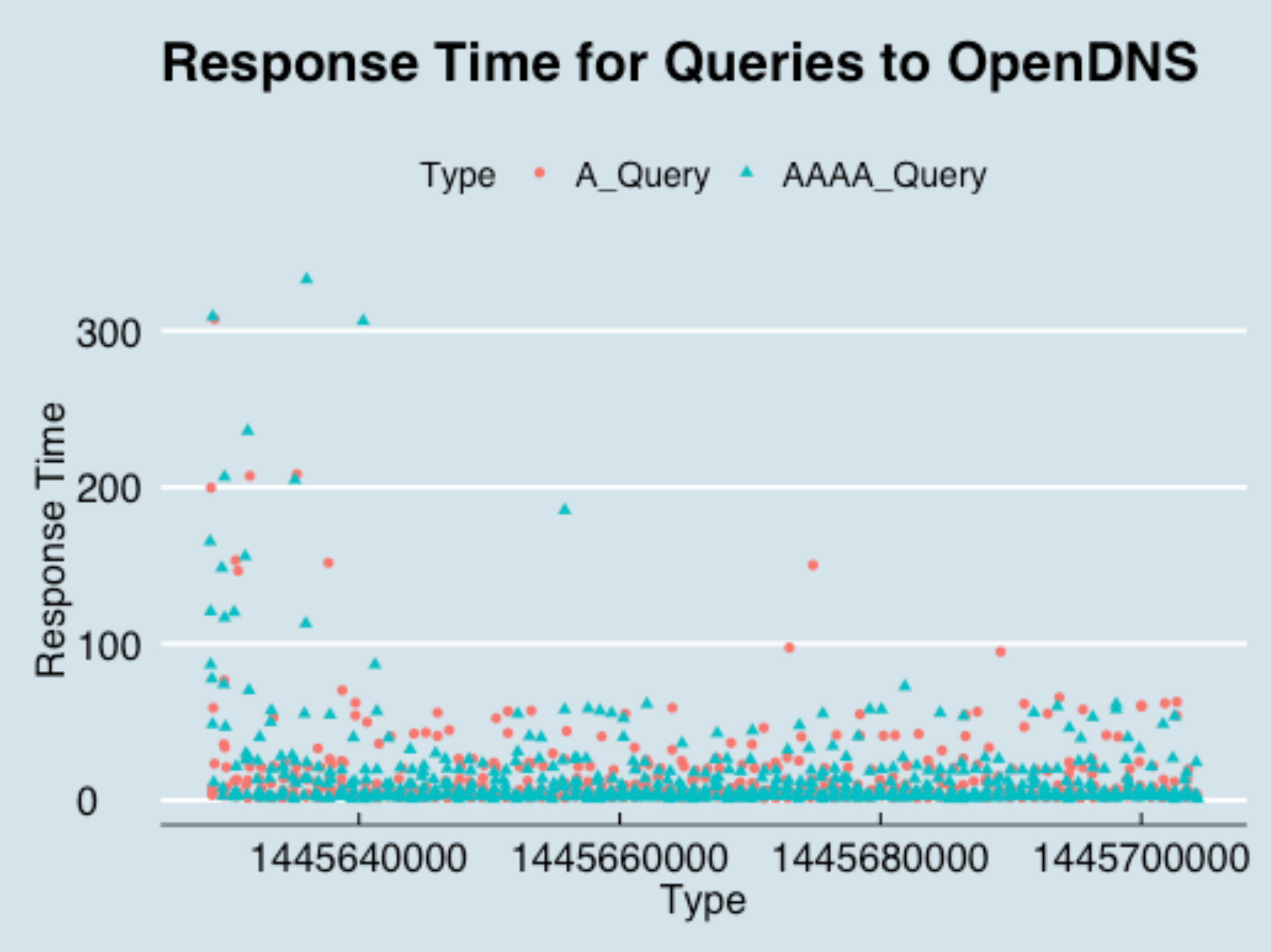


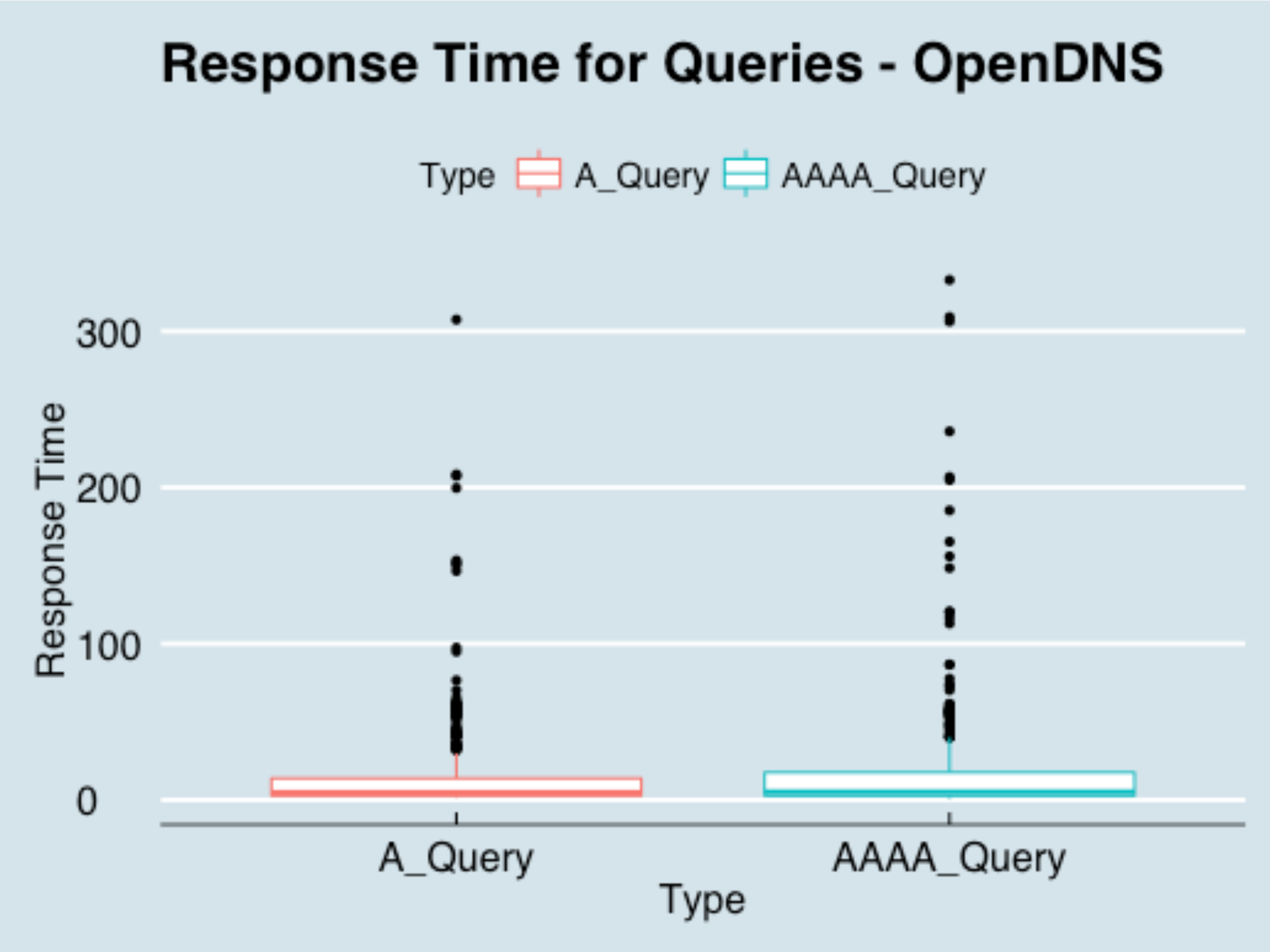
Ways of looking at the data

This set of measurements has a number variables of interest

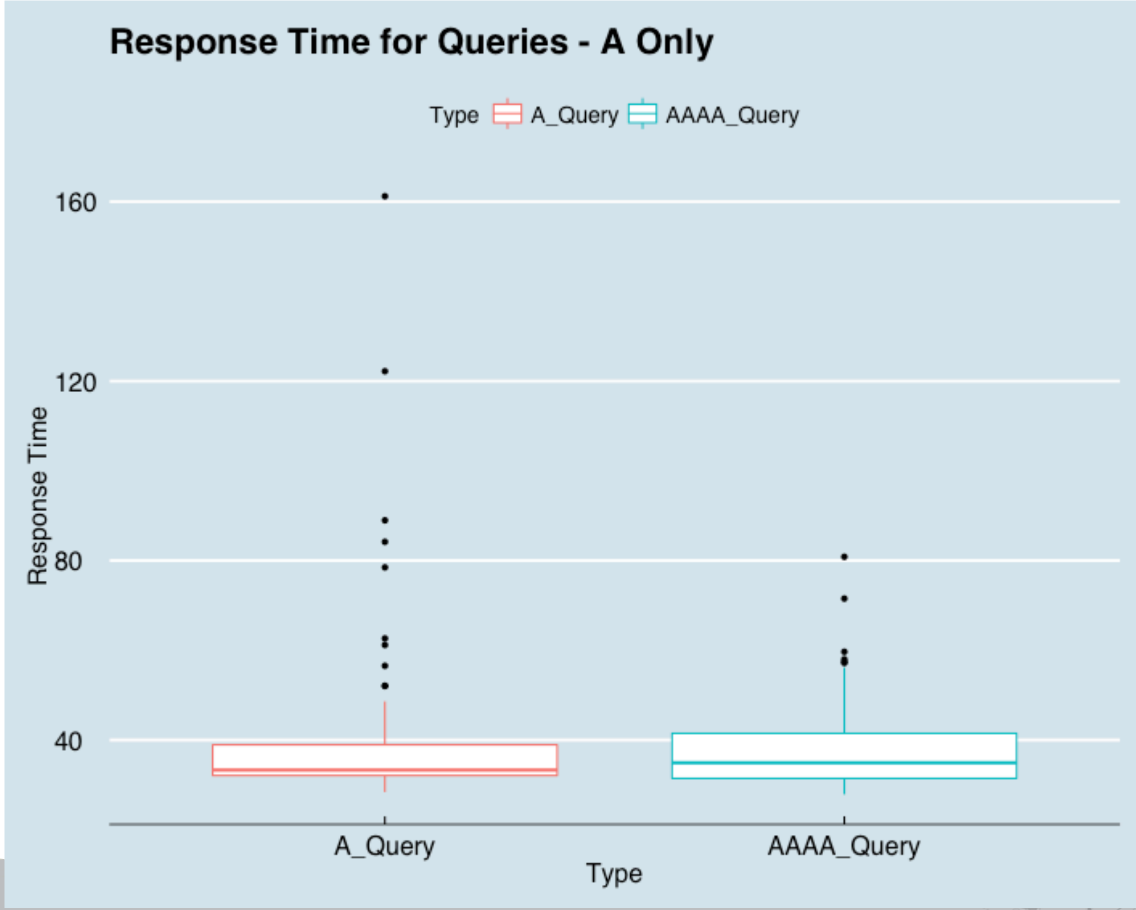
- **Destination Address**
  - Which resolver was this Atlas probe using?
- **Response Time**
  - How long did it take to respond?
- **Resource Request**
  - Was the measurement for an A or a AAAA
- **ASN**
  - What network is this originating from

With this wide range of variables its important to avoid making comparisons across variables which might impact results.

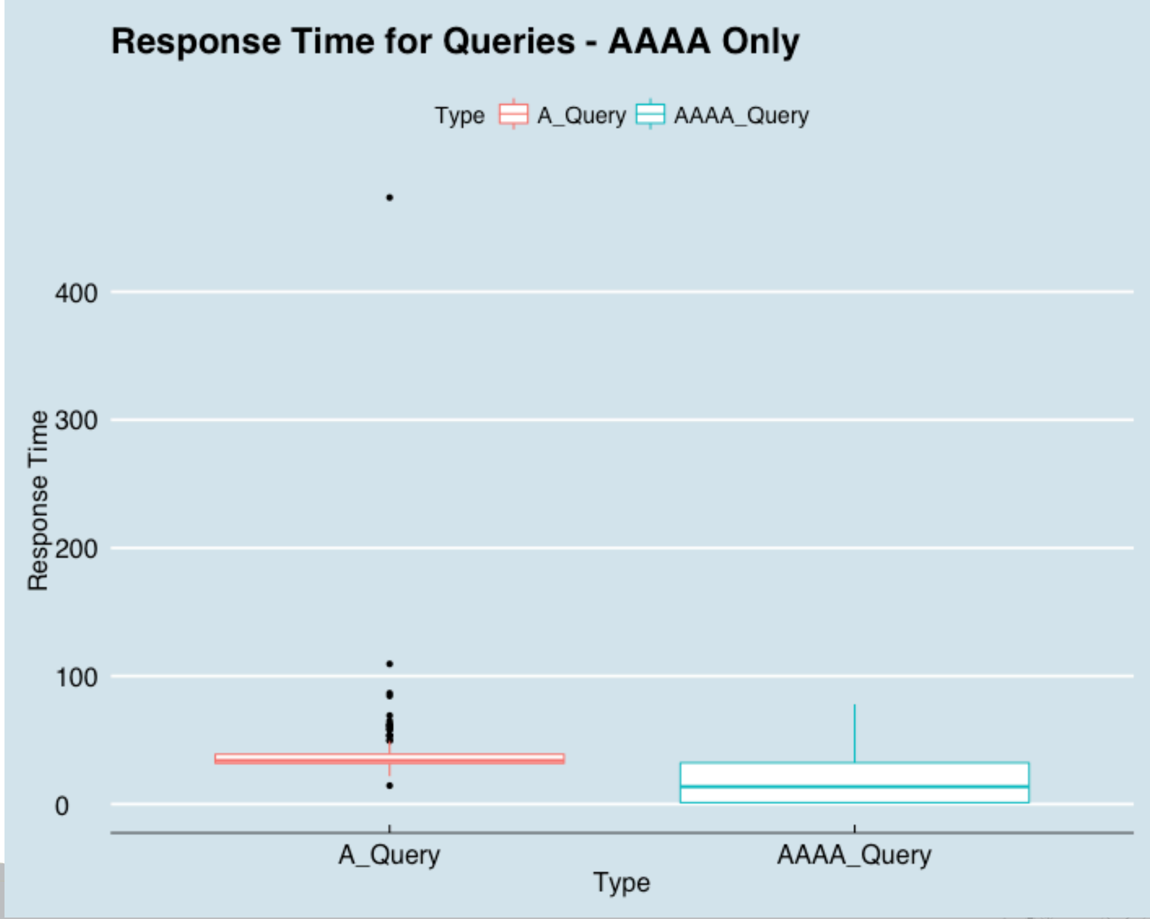

Response Time for Queries to OpenDNS
Type
A_Query
AAAA_Query
Response Time
300
200
100
0
1445640000
1445660000
1445680000
1445700000
Type


Response Time for Queries - OpenDNS
Type
A_Query
AAAA_Query
Response Time
300
200
100
0
A_Query
AAAA_Query
Type

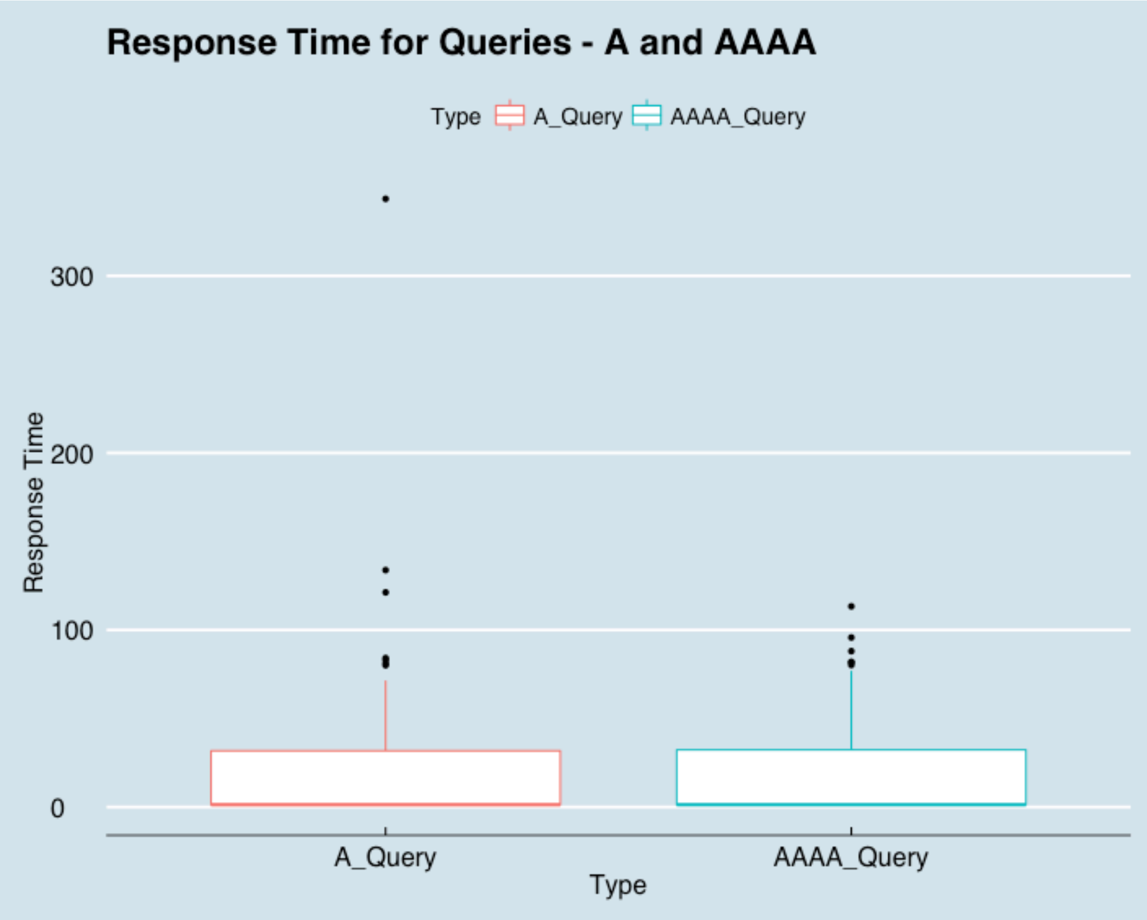
Looking At The Experience – Probe by Probe



Looking At The Experience – Probe by Probe



Looking At The Experience – Probe by Probe



Continued Study

When looking across the measurements by probe id there didn't seem to be a statistically significant difference in the response time between the request scenarios

- This negative finding to a pessimist means wild goose chase to an optimist it means your missing something ;)

Its important to consider that this dataset might require more research

- Learn more about the destination addresses and what they represent
  - Example: When the destination address is in RFC 1918 space do we need to think about the rate differently? Or think about the caching semantics differently?

As a follow up does it make sense to more explicitly test ISP and other open commercial resolvers instead of relying on the RIPE use local resolver?

Conclusions

Turns out IPv4 and IPv6 are different protocols and have a varying topologies

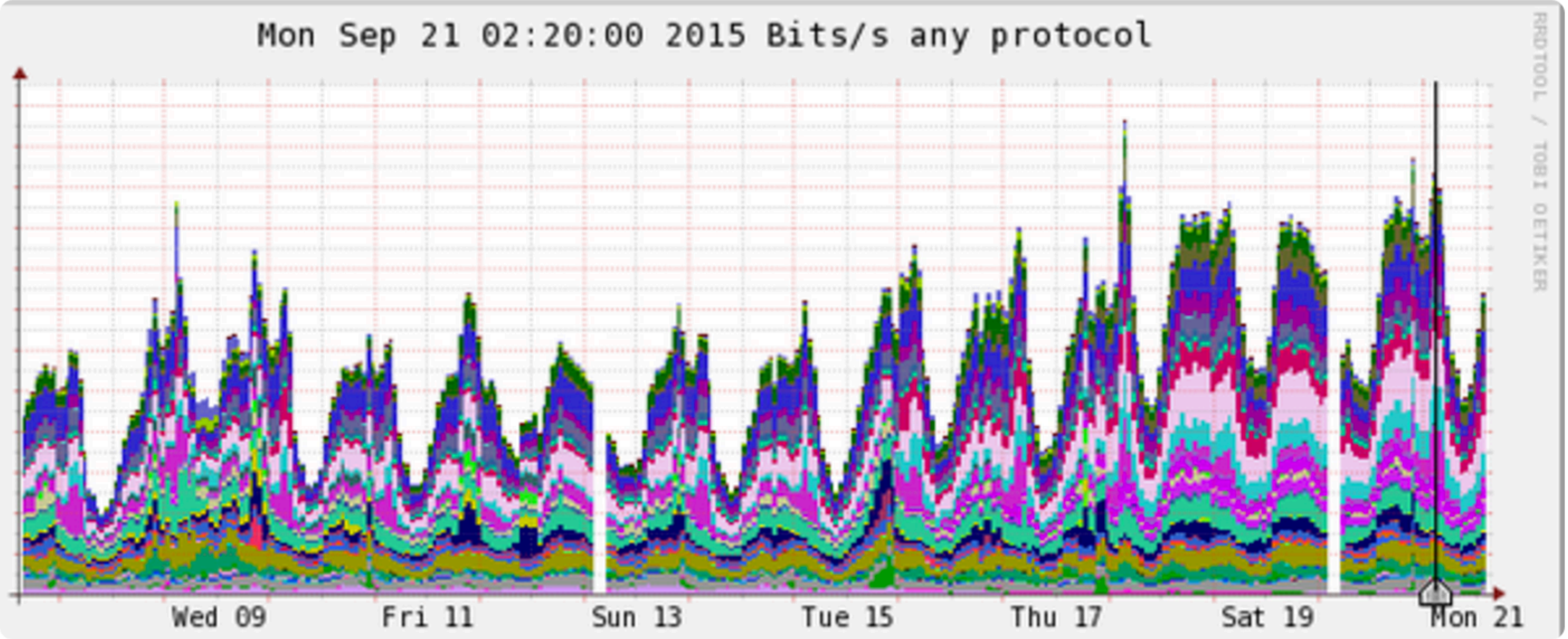
- Don’t assume parity between the protocols
- Ensure you are monitoring both independently

The 25ms delay, due to the timer waiting for AAAA responses, only impacts performance due to V6 routing … nothing to do with the DNS itself

- NO DATA responses were properly cached during testing

QUESTIONS?

Yes, iOS9 caused an #ipv6 traffic increase



https://twitter.com/jaredmauch/status/645968683794219013

THANK YOU!

Dyn[SM] INTERNET PERFORMANCE. DELIVERED.