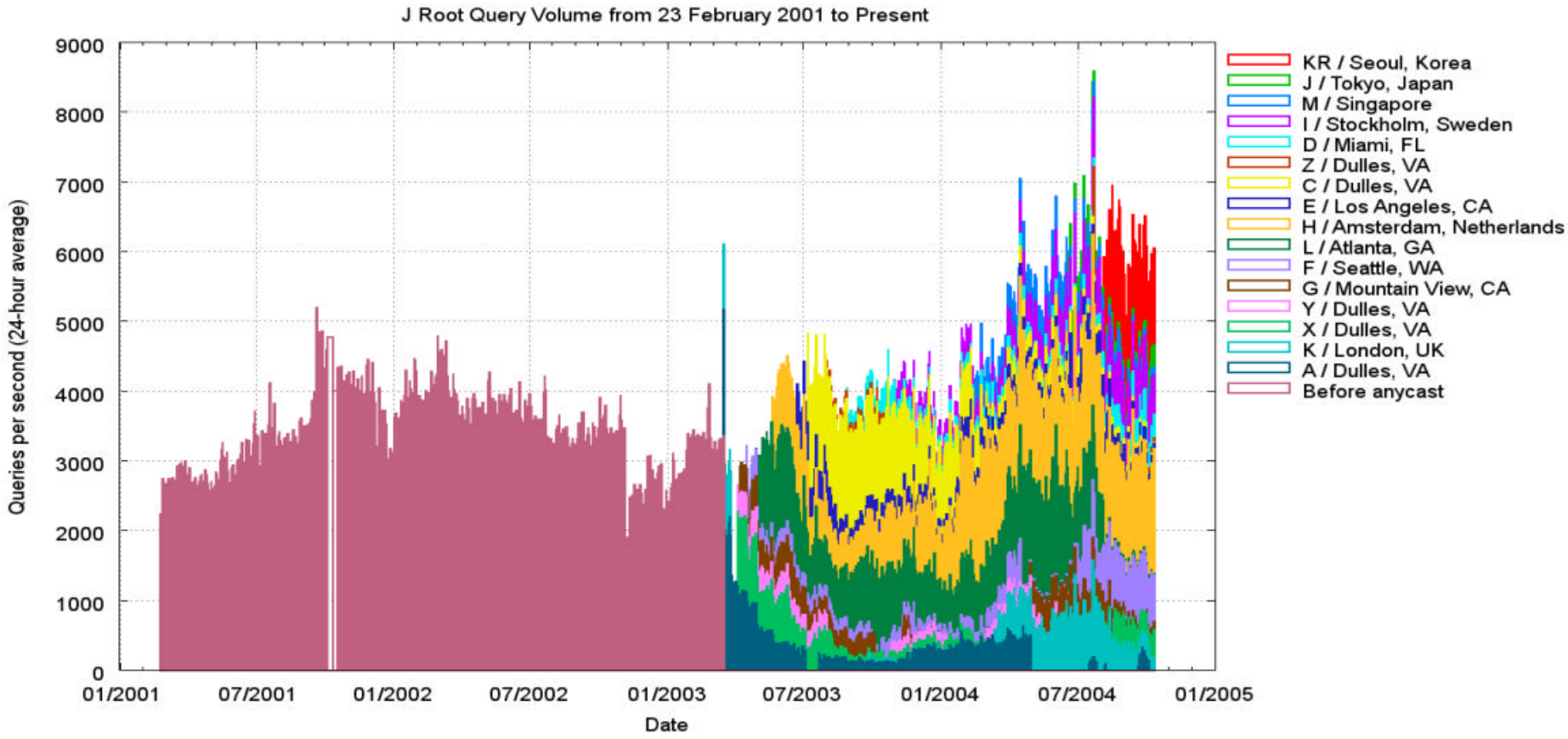
Life and Times of J-ROOT

Piet Barber
Matt Larson
Mark Kosters
Pete Toscano

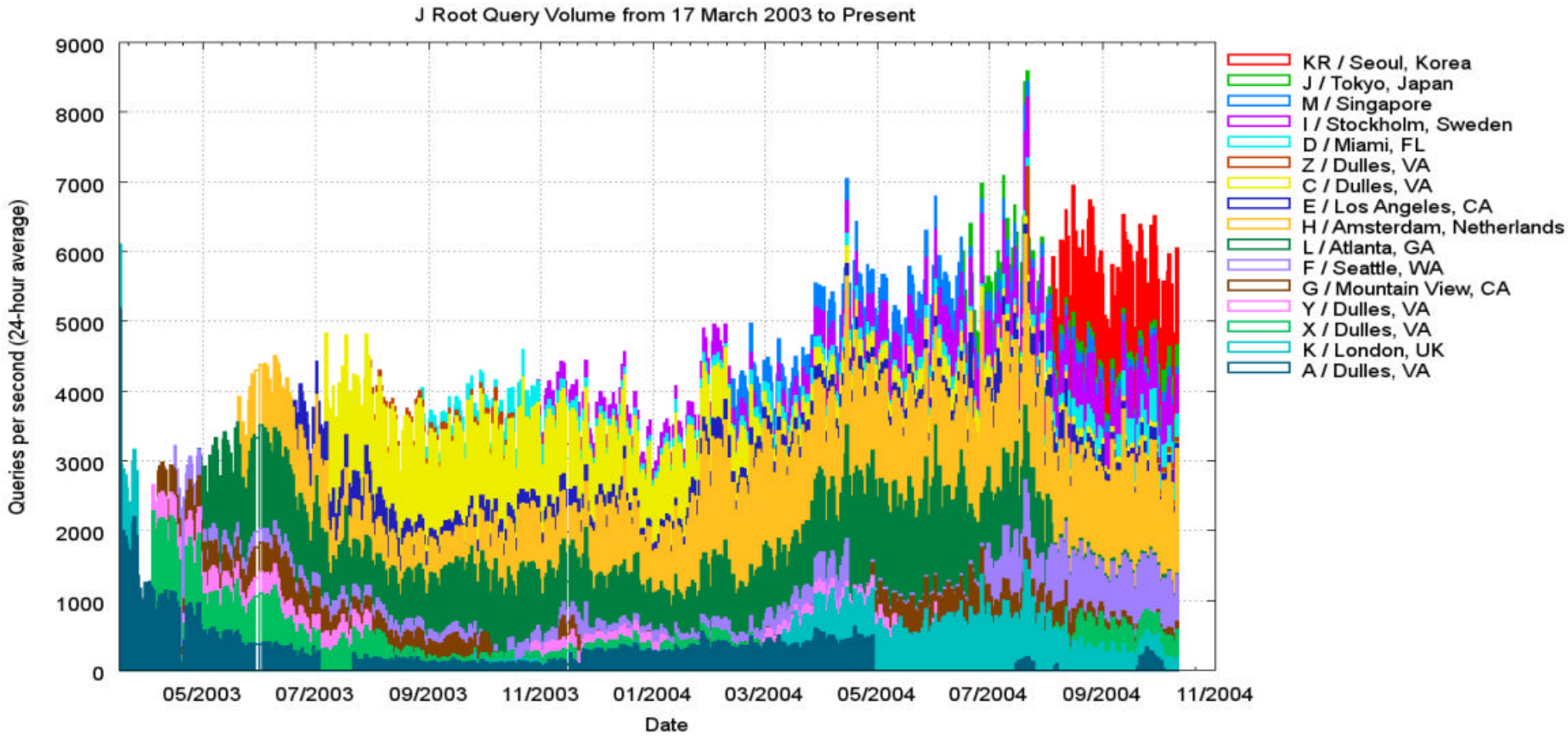
Agenda

- History of J-Root
- IP Address Change – Trends
- Move to Anycast

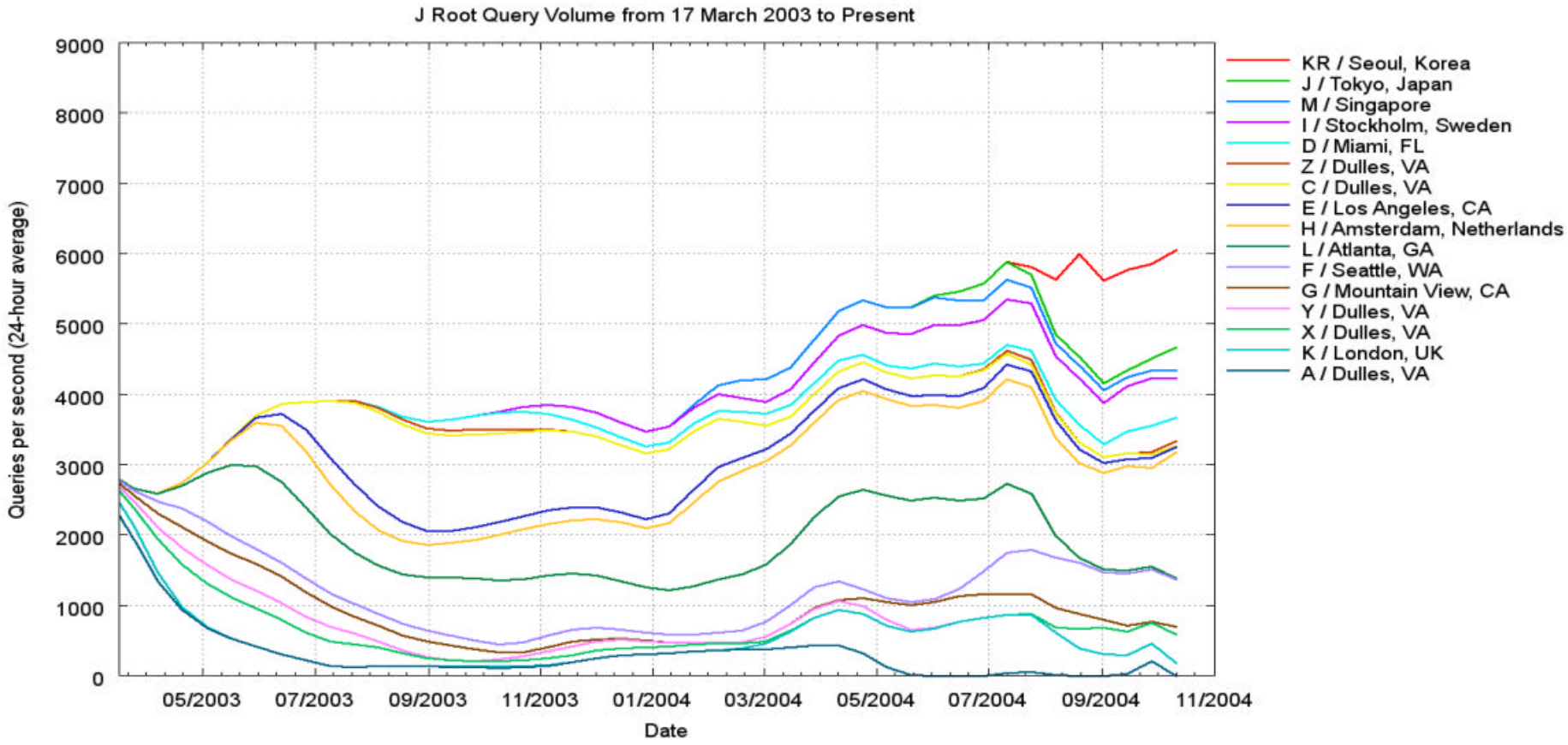
J-Root: Historical Query Volume (February 2001 to present)



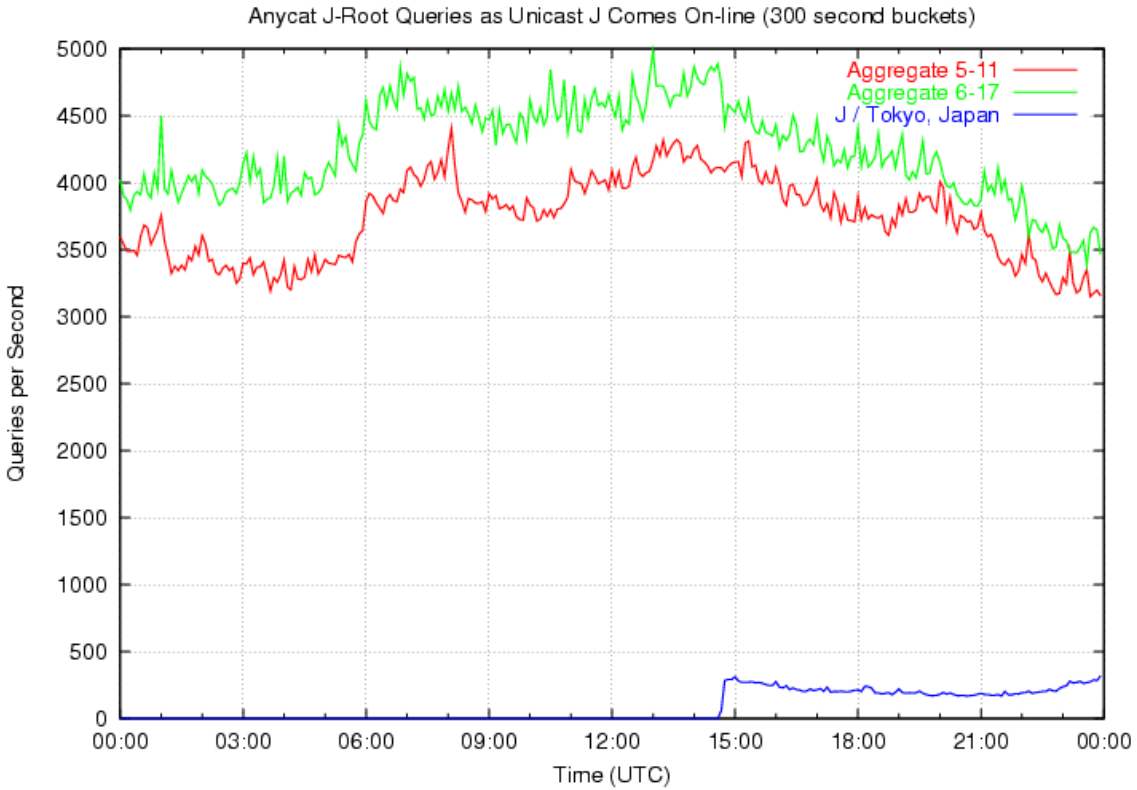
J-Root: Query Volume Since Anycasting (March 2003 to present)



J-Root: Query Volume Since Anycasting (March 2003 to present, alternate format)

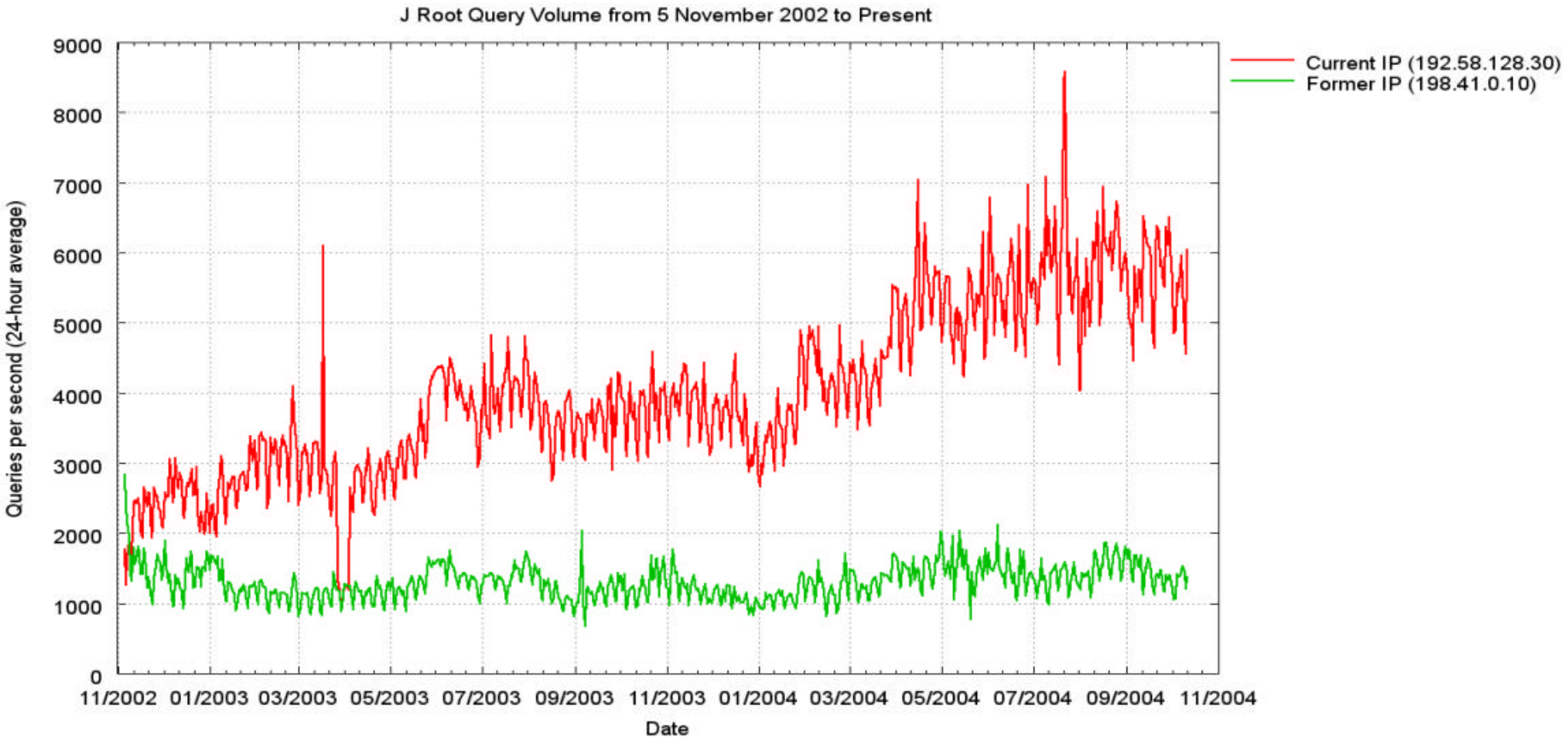


Traffic to a new instance of J in Japan

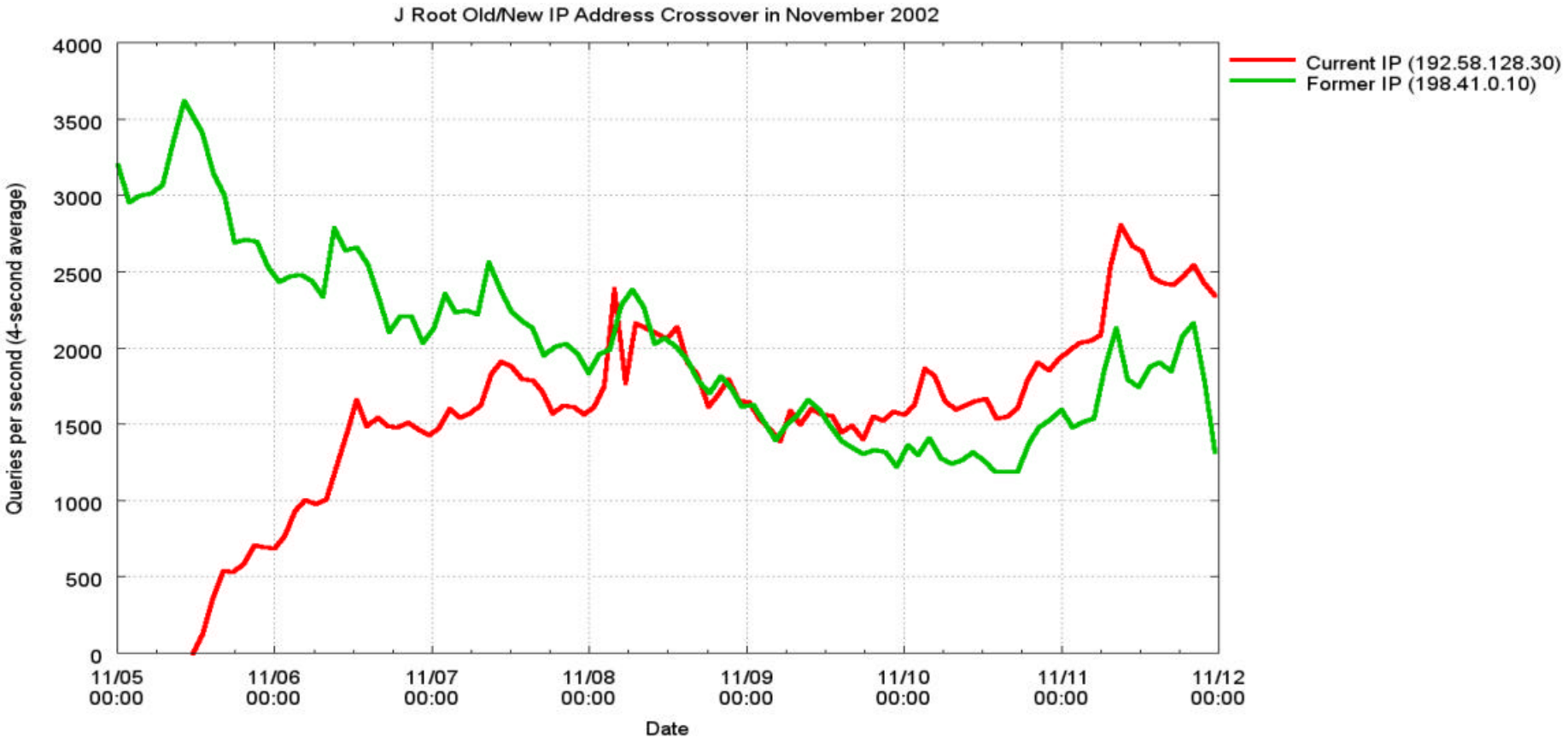


- New Instance did little to influence total (Green Line)
- Shows that roots have headroom

J-Root: Query Volume to Old/New IP Addresses



J-Root: Address Change Query Volume “Crossover”



Why Does the old J-root still have traffic?

- Who is querying the old J root IP address?
- Methodology:
  - Record a week's worth of source IP addresses
    - 6 October 2004 through 12 October 2004
  - Within each 24-hour period, retain only IPs that query the old J root **at least three times** (to rule out priming queries resulting from an out-of-date hints file)
    - 70,000-100,000 IPs per day
  - Aggregate all seven days' IPs and retain unique list
    - **205,307** unique IP addresses

Who is Querying the old J-Root?

- Used fpdns 0.9.1 to fingerprint all 205,307 addresses
  - http://www.rfc.se/fpdns
- As expected, large number of them were unreachable:
  - 139,927 timed out (68%)
- But those that were reachable proved to be a wide array of implementations
  - 141 different fpdns signatures/fingerprints
  - Details on next slide

Top 25 Implementations Seen at old J-Root

19453	BIND 9.2.0rc7 -- 9.2.2-P3 [recursion enabled]
10252	BIND 9.2.3rc1 -- 9.4.0a0 [recursion enabled]
9278	Microsoft Windows 2000
4893	Microsoft Windows NT4
4085	TinyDNS 1.05
2756	BIND 9.1.0 -- 9.1.3 [recursion enabled]
2651	BIND 9.2.0rc7 -- 9.2.2-P3
1998	BIND 9.2.3rc1 -- 9.4.0a0
1635	Microsoft Windows 2003
1350	BIND 8.3.0-RC1 -- 8.4.4 [recursion enabled]
898	BIND 9.2.0a1 -- 9.2.2-P3 [recursion enabled]
781	BIND 8.1-REL -- 8.2.1-T4B [recursion enabled]
603	BIND 9.2.0rc7 -- 9.2.2-P3 [recursion local]
602	BIND 4.9.3 -- 4.9.11
562	q0r?question section incomplete
479	q0tq0r?1,IQUERY,0,0,1,1,0,0,NOTIMP,0,0,0,0
229	BIND 9.1.0 -- 9.1.3
220	totd
216	BIND 8.3.0-RC1 -- 8.4.4 [recursion local]
209	q0r4q1r21q2r59q7r?connection failed
127	q0tq0r?1,IQUERY,0,0,1,0,0,0,FORMERR,1,0,0,0
120	q0tq0tq7r?1,QUERY,0,0,1,0,0,0,REFUSED,1,0,0,0
116	q0tq0tq7tq6r?1,UPDATE,0,0,0,1,0,0,NOERROR,1,0,0,0
109	incognito DNS Commander v2.3.1.1 -- 4.0.5.1
92	q0r?1,IQUERY,0,0,0,1,0,0,REFUSED,1,0,0,0

What's Going On?

- We don't know
- Old theory: Old J-Root gets traffic from implementations that don't prime
- Problem: Lots of recent BIND versions in that list, which are known (?) to prime correctly
- We need a new theory

J-Root Anycast Structure

- Each site globally visible behind AS 26415
- Other roots have different policies on anycast instances
- Local topology
  - Multiple boxes answering behind load balancers
  - Monitoring boxes sit in front of the load balancers

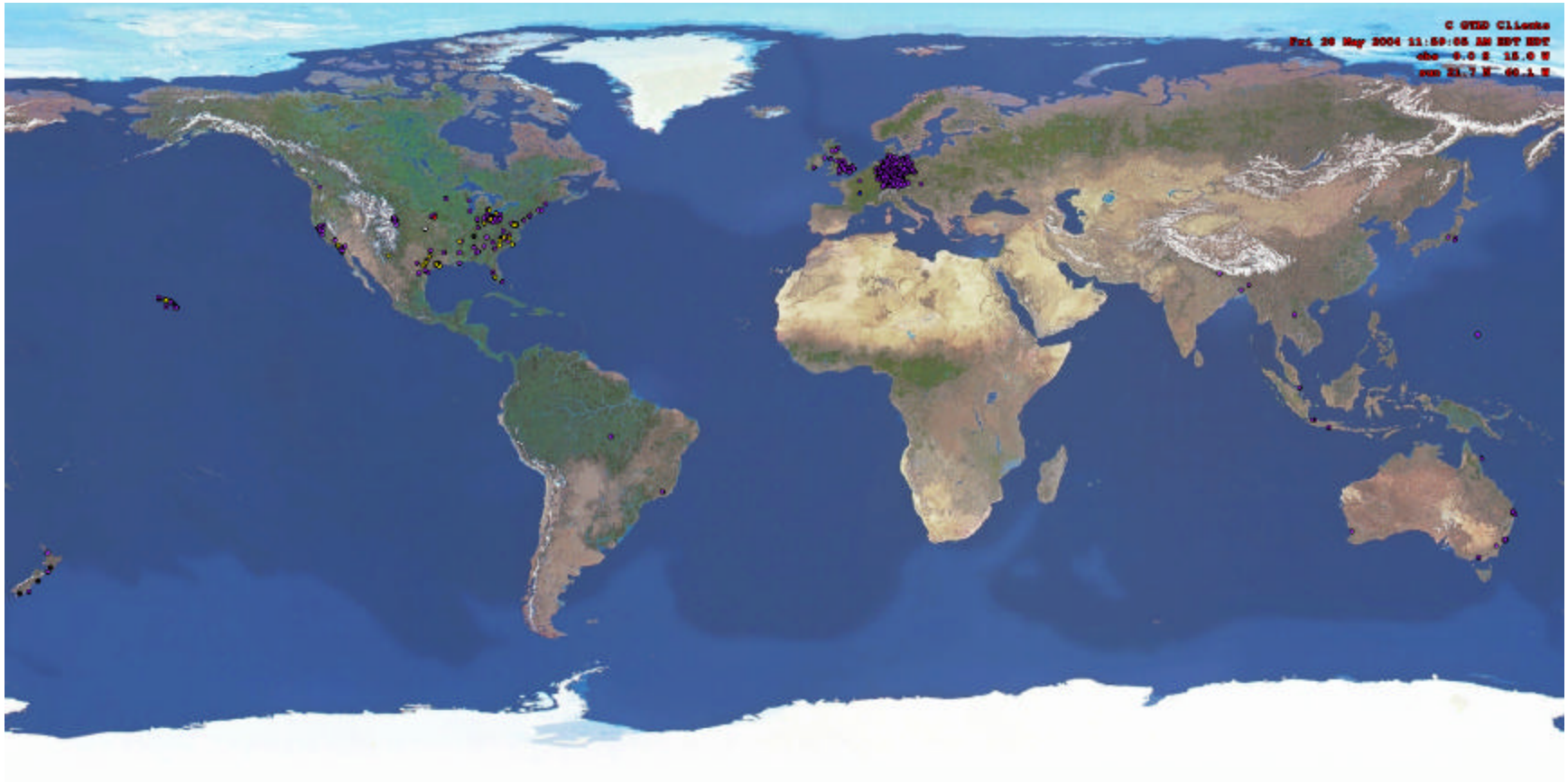
Analysis of Anycast Instances

- Look at Geo-mapping – good eye candy
- What types of systems are hitting these boxes
- What systems are asking for invalid TLD
- Interesting Behavior

Geo Mapping

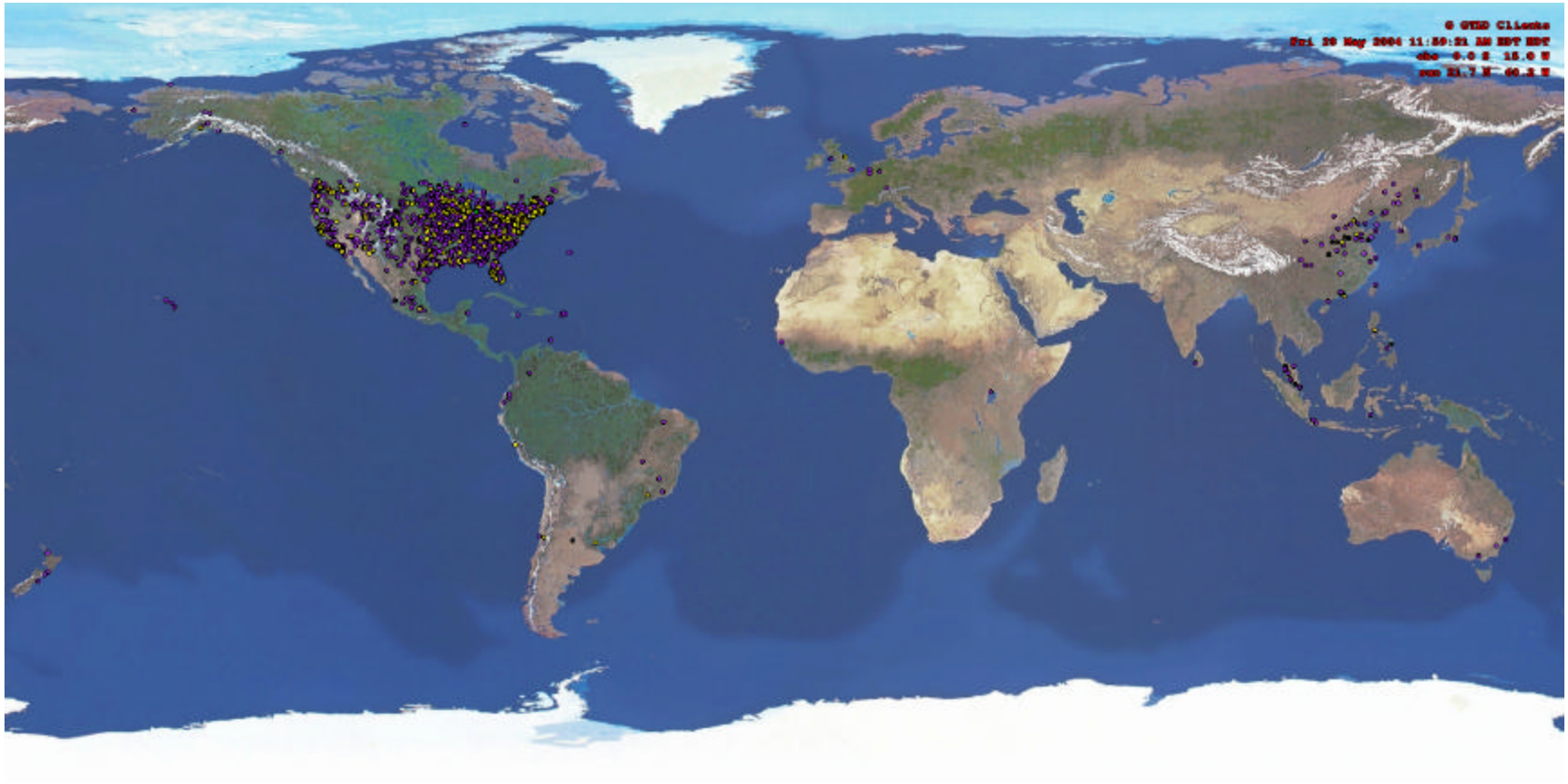
- Geo-mapped the sources and see if correlated to “physical site” location
- Colors depict # packets generated over time
  - Colors change over standard deviation with mean between red and green
  - White (highest)
  - Yellow
  - Orange
  - Red
  - Green
  - Blue
  - Purple
  - Black (lowest)

Geo Mapping (C/Dulles, VA)

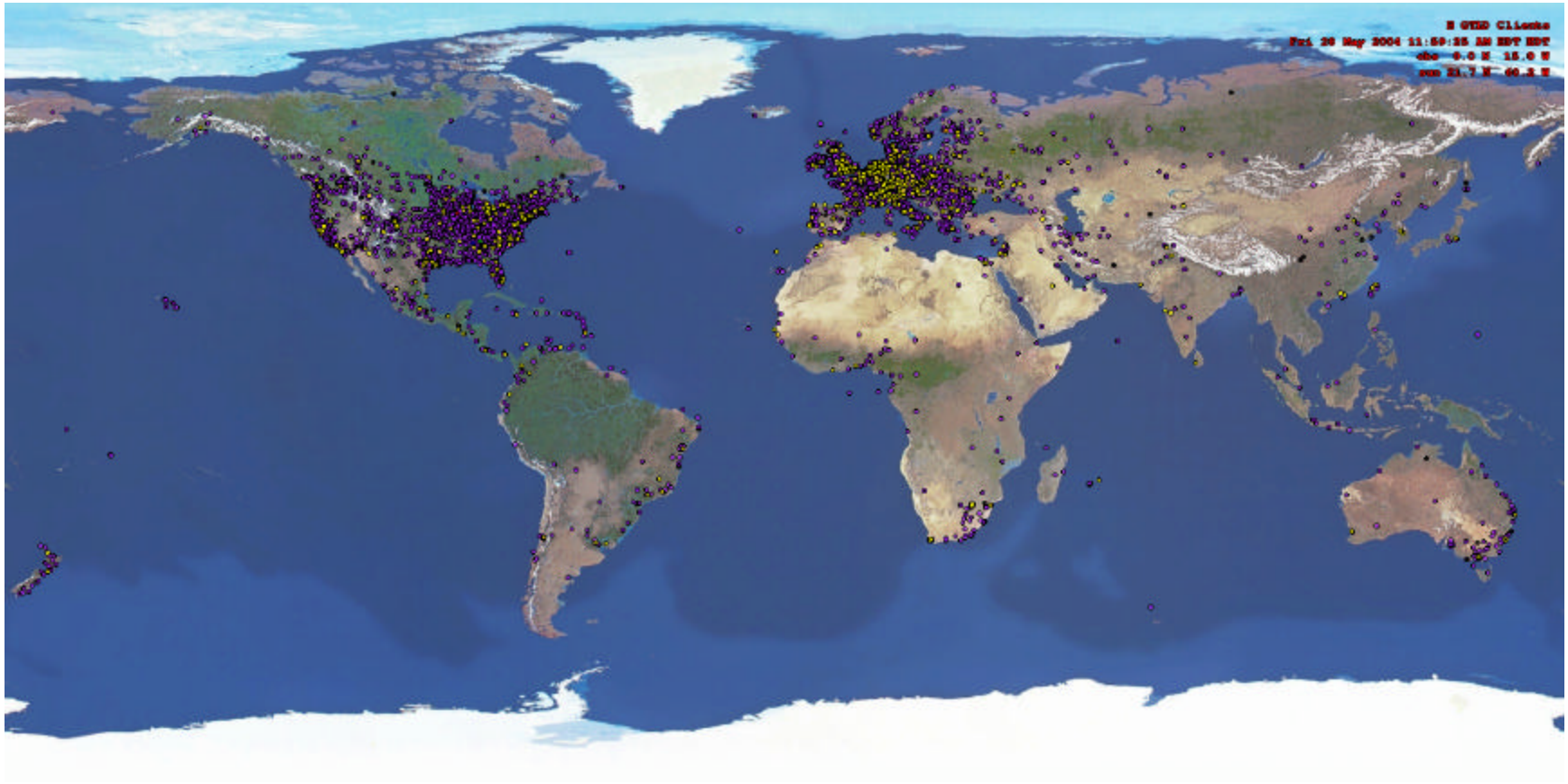


Geo Mapping (E/Los Angeles, CA)

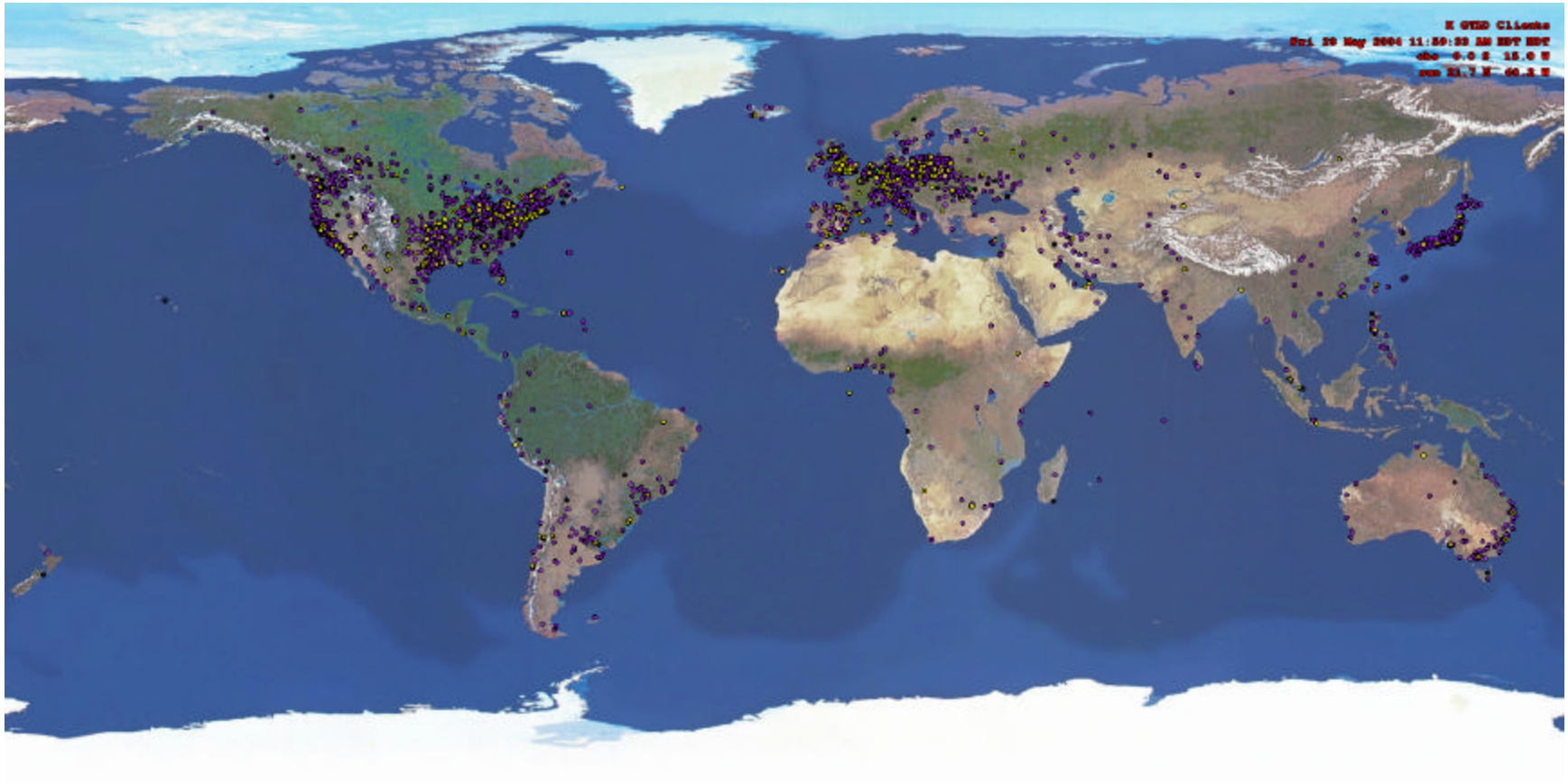
Geo Mapping (G/Mountain View CA)



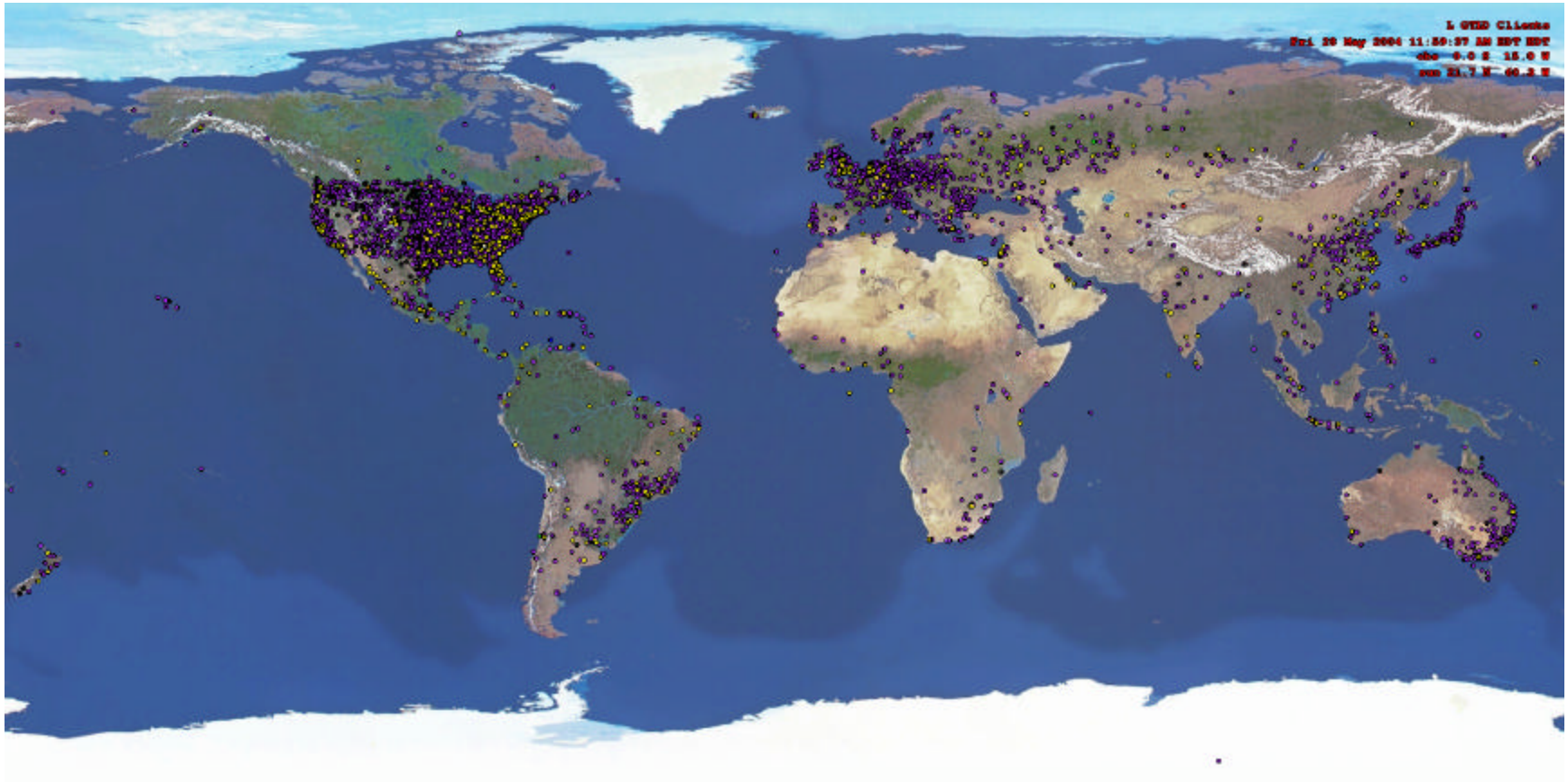
Geo Mapping (H/Amsterdam, NL)



Geo Mapping (K/London, UK)



Geo Mapping (L/Atlanta, GA)



Geo Mapping (M/Singapore)

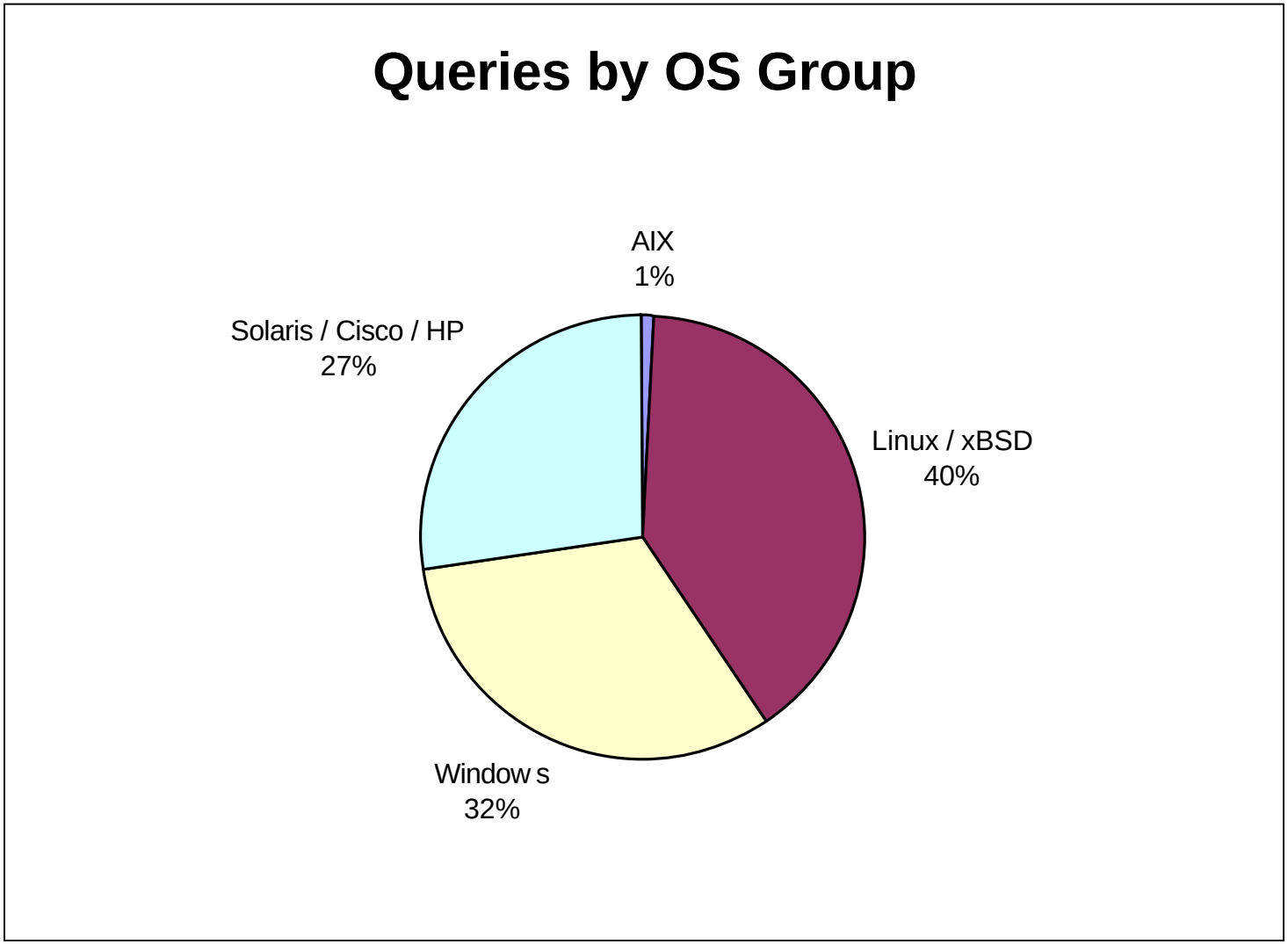
Graphs look nice but what does it mean?

- Used to be routing Location != Topology
- As Internet grows, location is becoming more aligned with topology

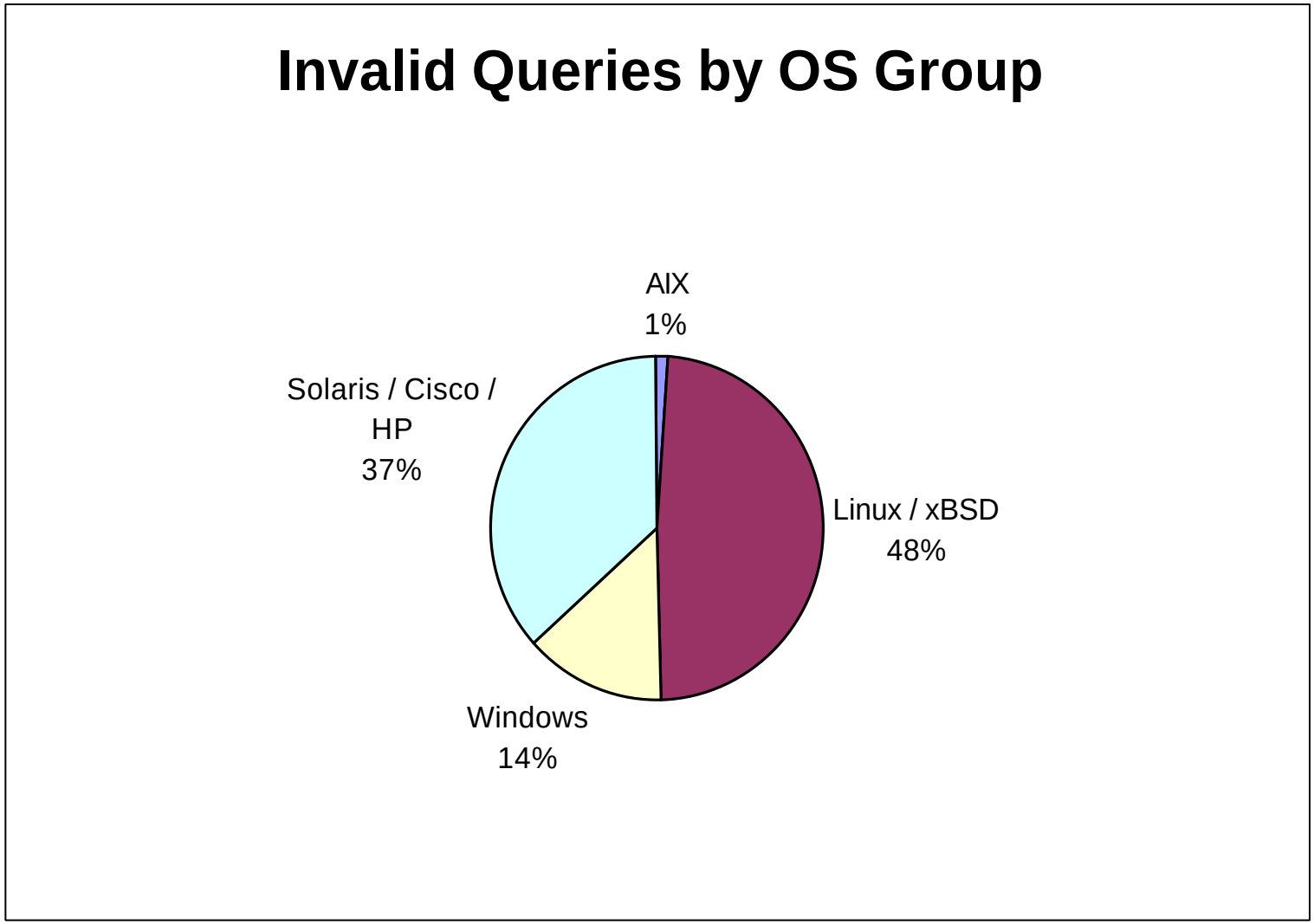
Collected TTLs on IP packets

+ Gives a “rough” correlation on OS that sent the packet
+ < 30 AIX
+ < 64 Linux/BSD
+ < 128 Windows
+ < 255 Solaris/Cisco/HP

OS mapping


Queries by OS Group
AIX
1%
Solaris / Cisco / HP
27%
Linux / xBSD
40%
Window s
32%

OS Mapping to RCODE 3 responses

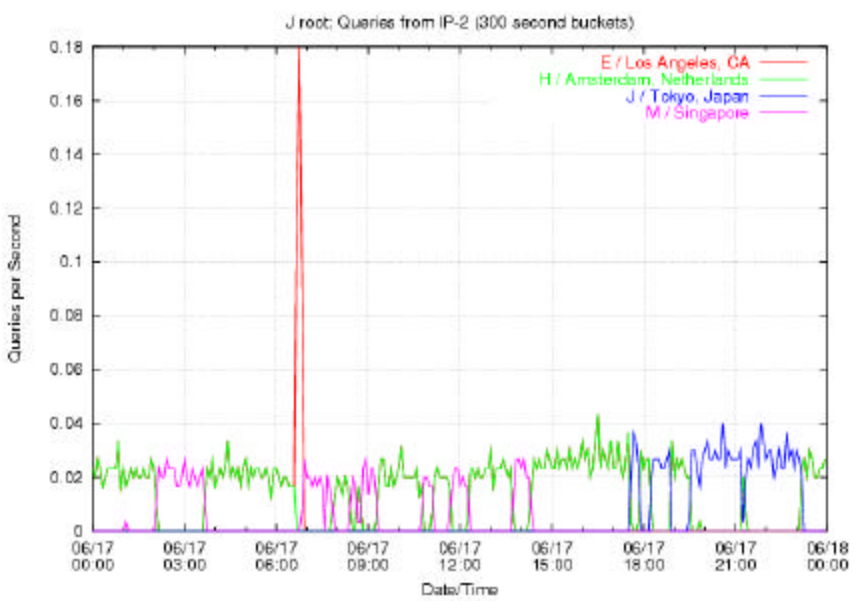


Interesting Behaviors

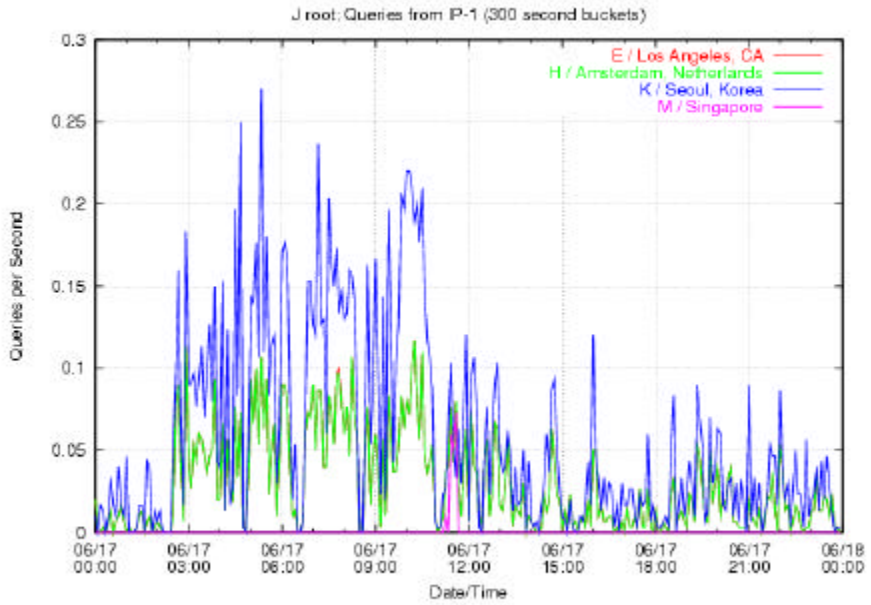
- Looked at sites that were seen in two or more anycast sites
- 3.69% of all traffic over three days was seen at two or more sites
- More than expected
- Looked at a couple of examples…

Traffic from one IP seen at multiple sites

"Normal Jitter"



"Abnormal Jitter"



Take home

- Expected to see a saw tooth distribution – instead have a noisy distribution in many cases
- Does not affect UDP
- DO NOT RUN Anycast with Stateful Transport
  - Will “No Export” mitigate this behavior?

Conclusions

- Just In Time Presentation
  - Very little work done with others
  - Need to work with other roots and core routing people
- Retired J Root
  - Reason for continued steady stream of traffic is unknown
- Anycast
  - Roots have multiple ways of doing anycasting
  - Questions that come to mind
    - Is one way better than others?
    - Is the diversity worth having some “suboptimal” configurations
    - Influence on IPv6 and DNSSEC that may escalate interactions into stateful transport