APNIC DNS Measurement & Perspectives on ‘DNS Health’

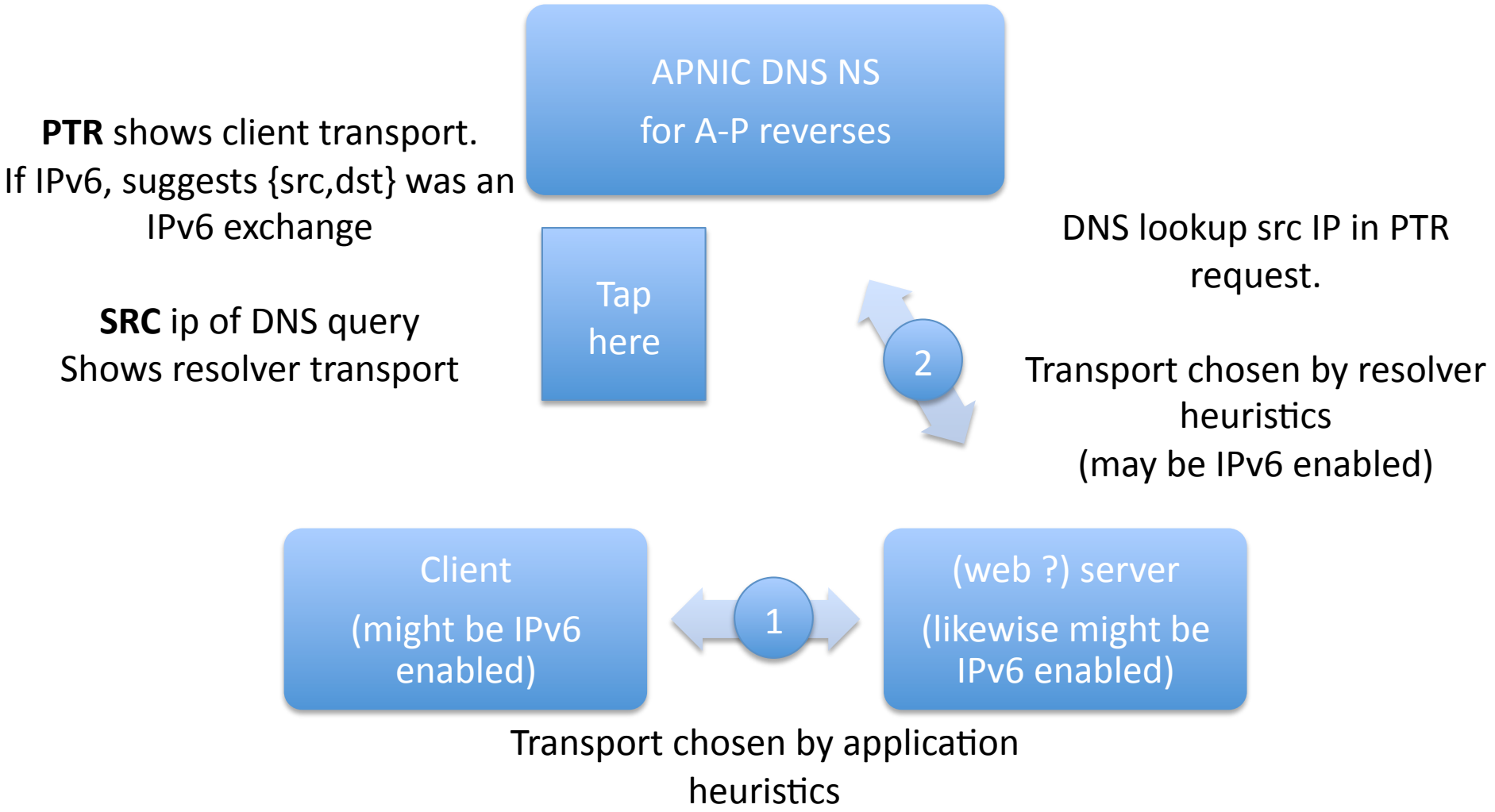
ggm@apnic.net

please, not ‘state of the art’

- What we do is not ‘state of the art’
  - It might be black art. At times it looks pretty artless.
- “organic” systems development
  - Accretion of habits, with occasional bursts of activity
  - Stick with what works, hack on what doesn’t
- We believe we could do better
  - Looking for clue, commonalities
  - Metrics?
- Therefore much of this slide pack is self criticism
  - “look on my works, ye mighty and despair”

We do reverse-dns.

Why we see your dialogues on the net (and what it is we see)



Reverse DNS

- Forward DNS is ***speculative***
  - "if I wanted to go somewhere, where can I go?"
    - ..but does the end-to-end dialogue ever happen?
- Reverse DNS is ***introspective***
  - "who was it who just talked to me?"
  - Highly structured namespace, few RR types
  - Potential for deep delegation
    - But its mostly 3 deep (the dots in v4 & in-addr.arpa)
  - Presumption that it mostly reflects {src,dst} pairings which took place.
    - Resolver modeled as belonging 'close' to the destination (server)

"its all just dns"

but some DNS is more 'systems level' than others

Why do people do reverse-DNS?

- …we're here because we're here (because…)
  - History.
    - Coded into SMTP server/spam filter logic
    - Coded into SSH daemons
  - Log file analysis ?
    - But should see clearer 'midnight' signals for log roll/processing
    - Overall load shape is diurnal (US swamps) but few economies show strong signals (JP exceptional)
- Emerging new uses
  - Geo Priv can leverage reverse-DNS
  - DNSSEC may add value: 'authoritative' name of address

Why indeed?

13 March 2009

apache and reverse DNS hostname lookups

No, no, no, it ain't me babe!

[digital]

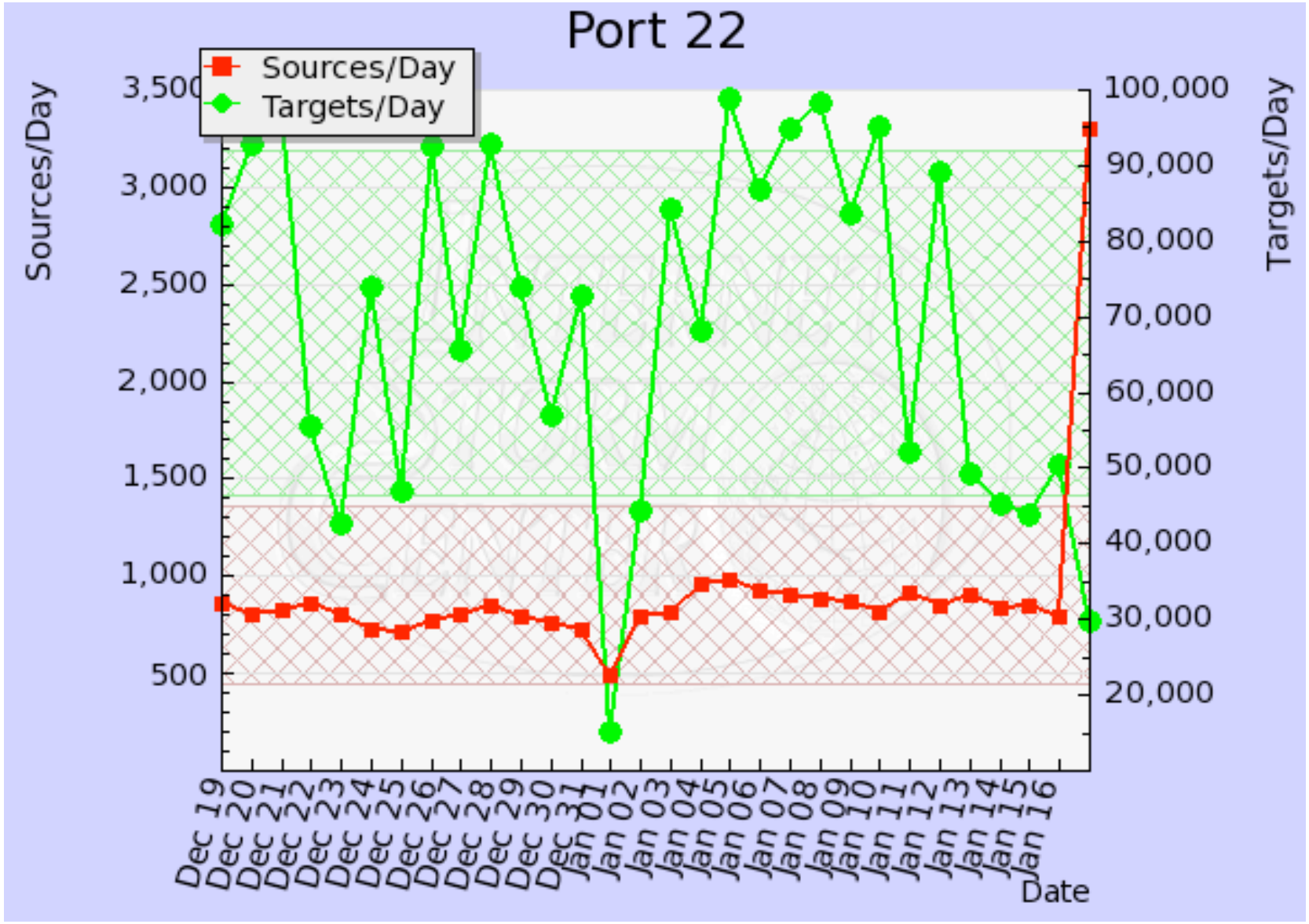
Doing reverse DNS hostname lookups on every request in apache (to have those domain names in the log file) is a bad idea as it will slow your server down, everybody seems to agree on that. Yesterday I noticed one of our servers doing just that, despite me not having remembered turning that feature on. Took me a bit of searching and cursing to find the culprit.

There are a few reasons why Apache (or Apache 2) will start looking up hostnames:

- `HostnameLookups on` somewhere
- checking of allow / deny rules with domains instead of IP ranges, e.g. `Allow from www.example.org` or `Deny from example.org`
- a rewrite rule with a condition like `RewriteCond %{REMOTE_HOST} www.example.com` (the last two I found on this thread)
- **update 2009-06-23** according to a post on a "simplywebhosting kb" , having `Deny from none` somewhere (this caused the problem to reappear for me one more time), apparently "none" is not a proper apache directive
- and the winner was: using `%h` in a LogFormat directive instead of `%a` (`%h` will give you the hostname, no matter what `HostnameLookups` says, `%a` will give you the IP address)

... and now please don't ask me why Apple (or is it Apache 2 at fault?) has the `LogFormat` with `%h` in their config on Mac OS X 10.5.

Logging bad guys.. (dshield.org)



Why do people do reverse-DNS?

- The truth is, I don't really know any more.
  - "a fertile area of study" (I'd **like** to know)
  - What % of internet traffic causes reverse lookup?
- "Organic", perhaps not deliberate?
  - Left in older code, not added to newer code?
  - Now off by default in apache logging?
- For example in bittorrent clients as a value-add
  - Examine your peer set
- My fingers routinely type `tcpdump –n` now

DNS Monitoring/Measurement

APNIC uses Zenoss for systems monitoring, including DNS systems

- Zenoss is pretty good for base OS/systems view
  - Interface packet counts
  - I/O costs, Disk traffic/volumes, CPU time
  - Good graphing, consistent UI.
- Zenoss not so good for basic DNS monitoring
  - “Check port 53 answer status” is about it
  - Hand-scripted in-addr.arpa specific zone queries
    - Zone serial, zone size checks. (heuristics)
  - Overly simplistic, but we’re still learning how
  - NMS has trigger concept, we should use it

Zenoss because …

- Because we ran nagios and upgraded (organic…)
- APNIC runs general purpose computing systems, web WHOIS mail & ftp and needed an integrated system for its overall OS/Network view.
  - Provides graphing and SMS/alarm framework
- The DNS monitoring in Zenoss is a retrofit.
  - Its not integral, or necessarily monitoring what really matters.
- More work needed here
  - First class event monitoring/SNMP/probe logic into commodity reporting/NMS
  - Triggers. Some rising waves easily detectable

DNS-specific systems monitoring

- 1 minute packet samples, 15 minute cycle (tcpdump) since 2002
- Full packet capture (dnscap) since 2007
- DSC for server/service specific monitoring

Packet samples

- 1 minute packet samples, 15 minute cycle (tcpdump) since 2002
  - Pre-DITL/DSC measures, maintained for backwards compatibility
  - Will be deprecated when relationship to other measurement confirmed, so far, good correlation to basic numbers
  - Now regret not keeping samples.
- Sampling is viable long-term, offers data retention possibilities
  - More bang-per buck over time, if # records matters

Packet capture (dnscap)

- Full packet capture (dnscap) since 2007
  - Deployed for DITL, but left active 24/7
    - Used for IPv4/IPv6 measurements, inter-economy measures
    - Used for what-if and WTF analysis on-demand
  - Ad-hoc (scripted) analysis on demand
    - (eg NZ data shown later)
- Different Visualizations/Analysis being explored

DSC

- DSC for server/service specific monitoring
  - Not integrated into DNS service management (yet)
  - Has alerted staff to problems (see DNSSEC)
  - gratefully acknowledge work of
    - ISC/OARC/measurement-factory
- This is the workhorse. Most value lies here

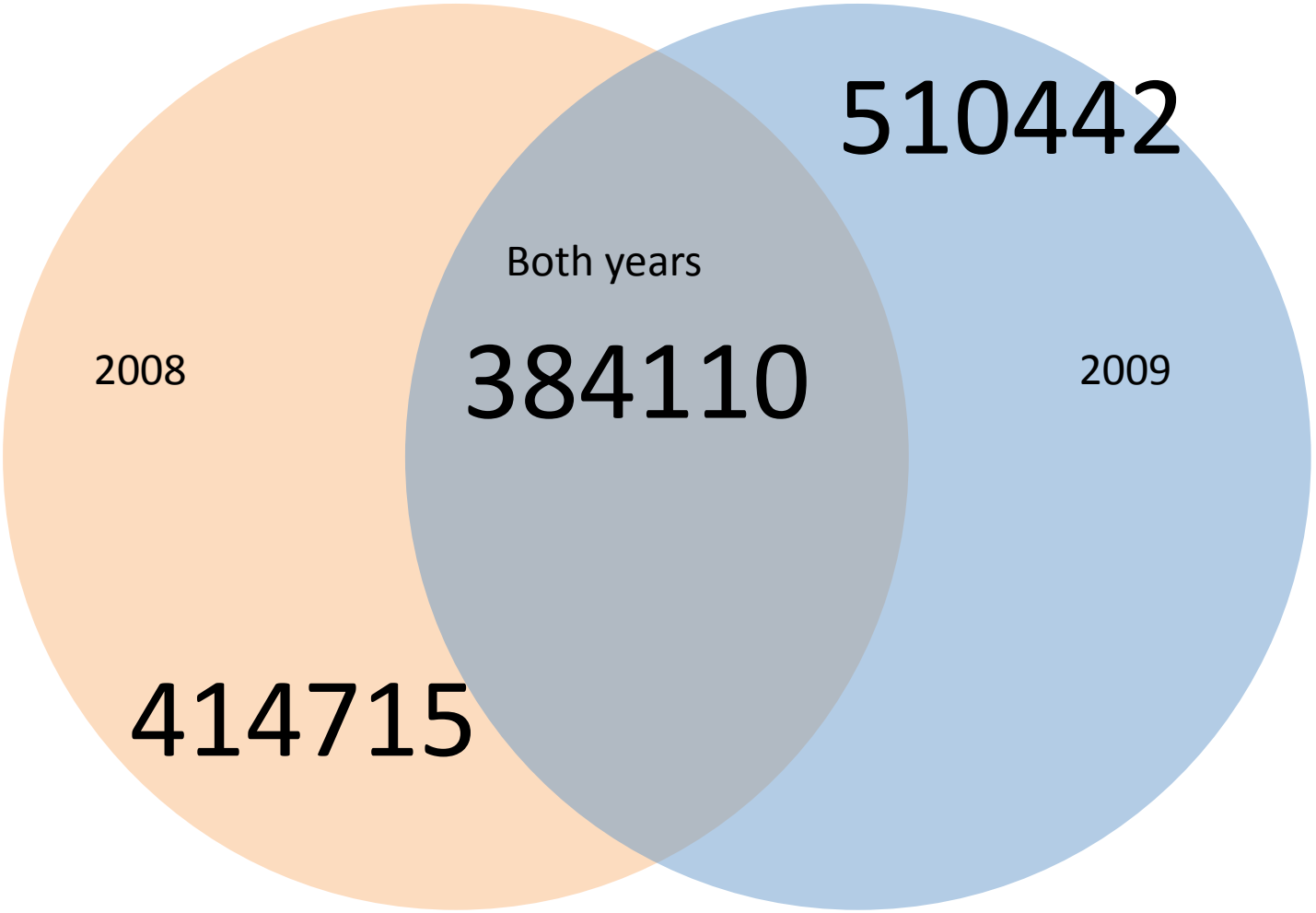
dnscap issues

- Focusing on the IP abstraction of DNS has ‘occluded’ other information from our minds
  - We forgot about ICMP. Router stats invaluable.
  - We aren’t adequately accounting for fragmentation, or cyclical/series behaviors from clients
    - This is fixable. We need to be more observant in general
- NCAP/PCAP confusion(s)
  - Do we really need ncap and not pcap?
  - What about the low(er) level issues, how do we measure fragmentation, ICMP …

Interesting behaviors from DITL

- The DNSCAP data is the basis of our DITL input
- What sorts of things are we noticing in the DITL data?
  - Only using own-packet capture at this time
  - Want to generalize, see if commonalities in other samples.
  - Also want to see if assumptions about 3 of 6 NS apply, ie get entire NS set for reverse into a DITL
- DITL is long-range, 10,000ft view crown-jewel
  - Looking forward to 2010! (the year of DNSSEC)

Unique IPs in 24h 2008-2009

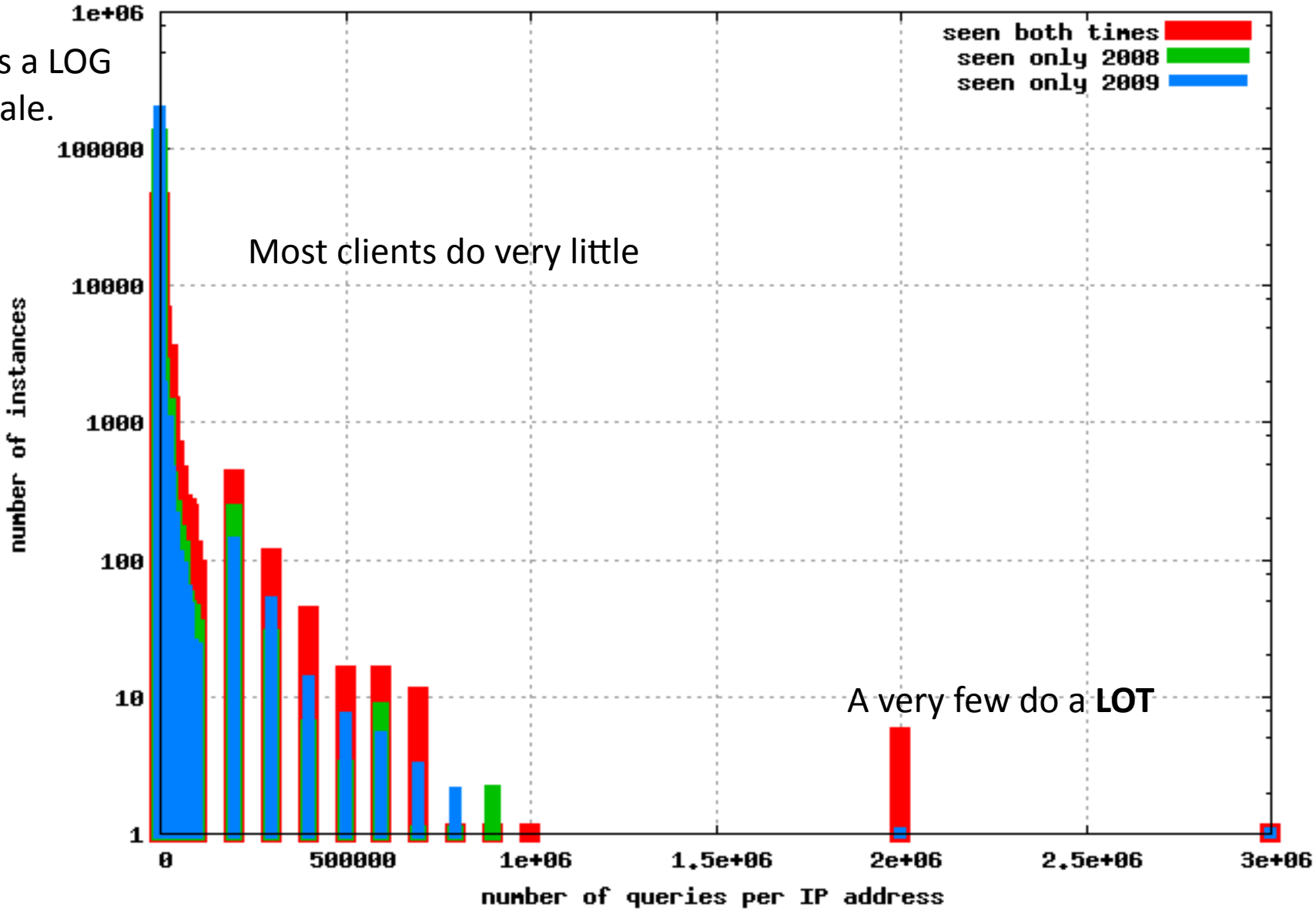


Address persistence in DNS 2008/2009

- < 1/3 overlap in addresses (re) used to query 1 year later
- Population of DNS query has stable and variant parts
  - Of the order 1.5m distinct addresses doing DNS lookup
- What is the relationship of this subset to the root 5m addresses?
  - (from Sebastian Castro DITL analysis slides, NZNOG)
- (this may be reverse-DNS specific behavior)

How often do people query?

It's a LOG Scale.



How often do people query?

- Large population of ‘occasional’ query src
  - Little persistent query, 1-2 hits, then gone
  - Mentally modeled as ‘not forwarders’ for now
    - Passive classification? What clues are in the query pkt?
- Small population of ‘continuous’ query src
  - Mentally modeled as ‘forwarders’
    - Should be able to see caching behaviors here
- More work needed here

IP transport selection is random?

- Emergence of IPv6 transport use by DNS query sources
  - AAAA transport querying for A PTR and vice-versa
- Transport selection algorithm in resolver is not understood.
  - (by me I mean. I'm sure people here know)
  - Is this 'accidental' V6 usage, or deliberate?
- 6to4, native IPv6 both seen
  - 6to4 has to be very sub-optimal compared to V4
  - No routine DNS (yet) is single-stack IPv6 only NS
- Maybe this is a good thing? Dual-stack emerging?

Who uses IPv6 for DNS transport? A DITL-like look at NZ in 24h (from DSC)

- Last week, What % of NZ dns queries use IPv6 transport?
  - 0.65%. ~ one in two hundred
  - This is comparable to other measures APNIC does of IPv6 transport in DNS, worldwide.

- Last week, What % of reverse-DNS was for ip6.arpa or in-addr.arpa?
  - 0.01%. One in ten thousand was for ip6.arpa.
  - That's the % of NZ addresses queried that are IPv6.

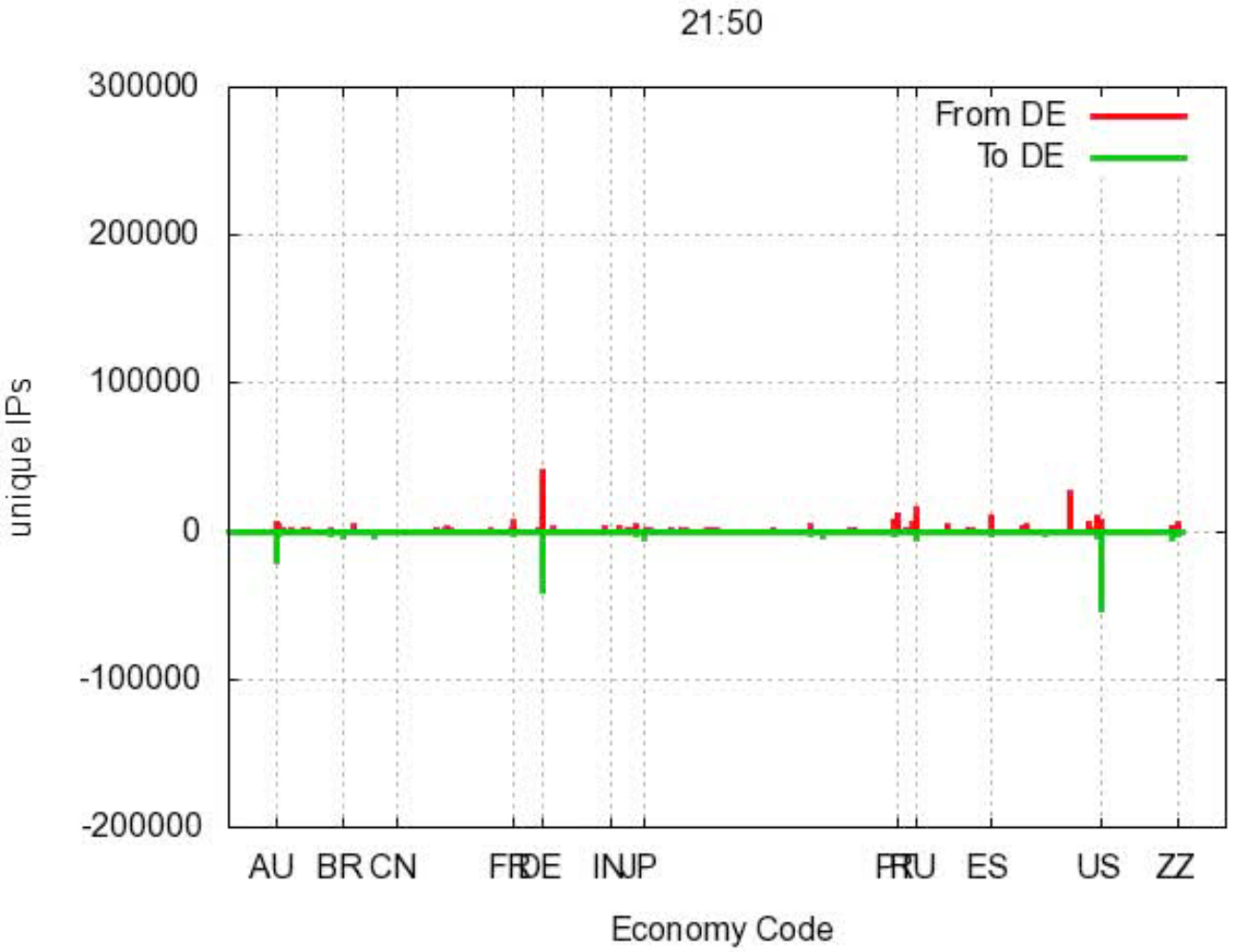
Who uses IPv6 for DNS transport? A DITL-like look at NZ in 24h (from DSC)

- Yes, more DNS flows **over** Ipv6, than is questions **about** IPv6!
  - The resolvers selection algorithm for DNS transport is 'sticky' compared to random client IP transport selection algorithm in the browser. Browser strongly de-preferences V6
  - Any NS answering is acceptable. If it answers, use it and stick to it.
- V6 transport for V4 PTR, V4 transport for V6 PTR..
  - It's a dual-stack world.
  - (from NZNOG presentation)

Who used IP to get anywhere? A DITL-like look at NZ in 24h (from DSC)

- In IPv4 1080 allocations visible, from 1234 marked to NZ in delegated stats file
  - 87.5%
  - 1.6m distinct IP looked up as PTR queries.
  - That's pretty high: many economies see less visible use of their IP ranges in global DNS
- But for IPv6 only 21 allocations visible from 46
  - 43%
  - 143 distinct IPv6 looked up as ip6.arpa PTR queries
    - (from nznog presentation)
- This is probably worth collating/tabulating for all economies as a standing report on address usage.
  - Comparable to other OECD data collection inputs

Inter-economy flows in reverse



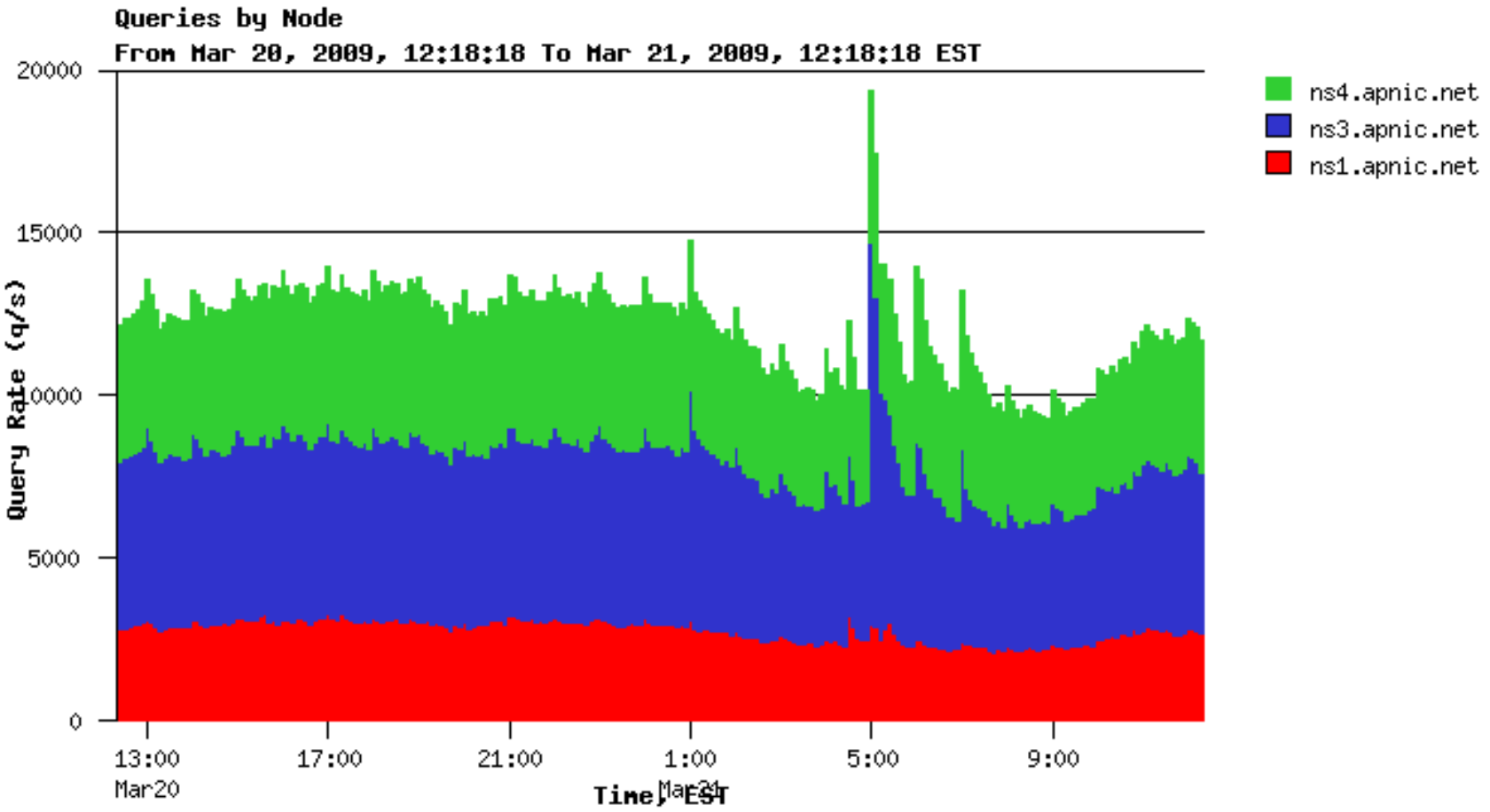
Inter-economy flows in reverse

- Continuing question from Geoff
  - "yes George, but what does it all **MEAN**"
  - I wish I knew.
- **Intra**-economy signal
  - Strong .de .de signal
  - All economies show this. The size of the peak varies massively
- **Inter**-economy signal
  - Why does .de query about .in IP ranges?

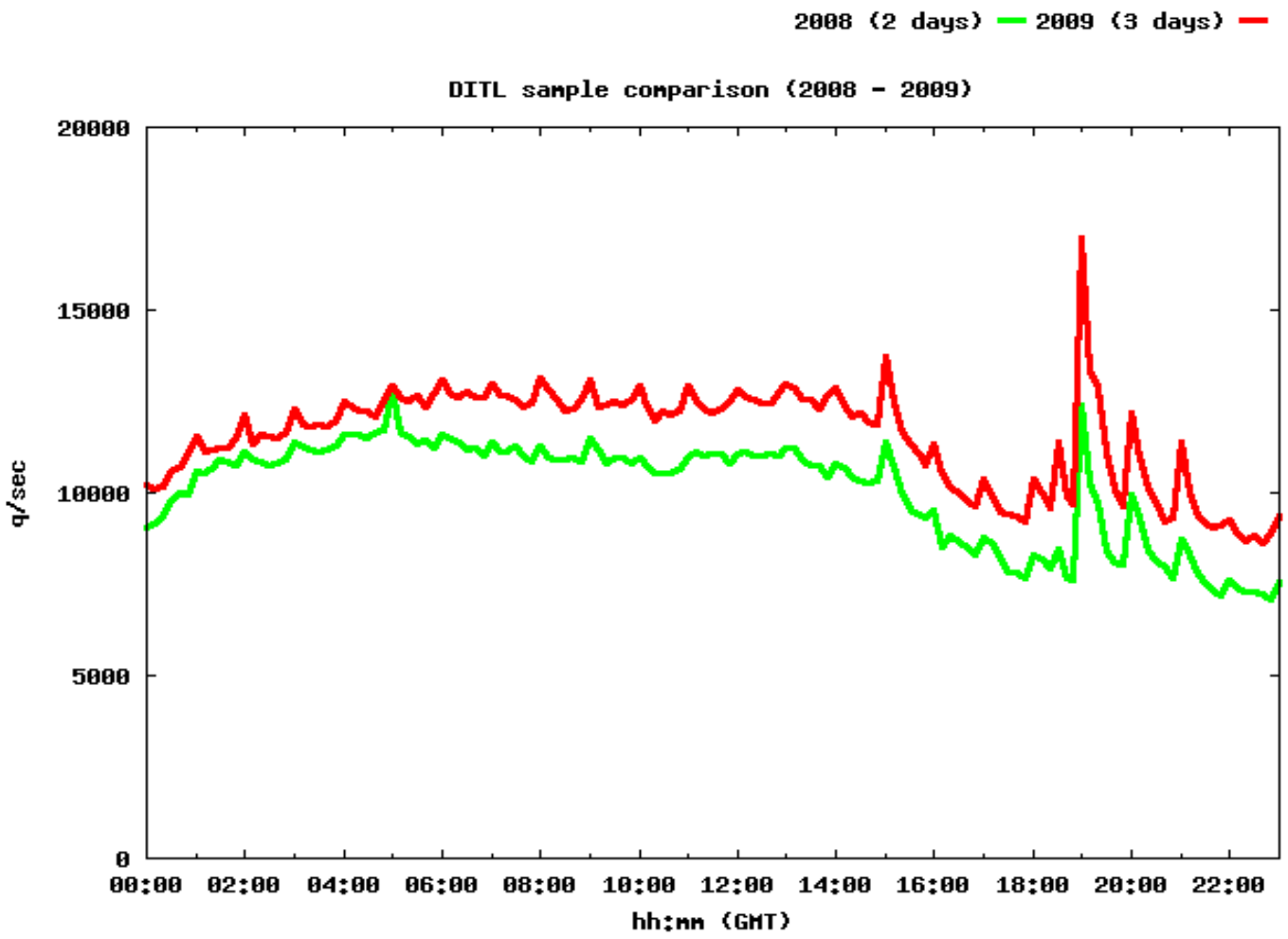
DSC

- Has worked extremely well for us with only one caveat:
  - In-addr is not considered an ‘interesting’ top domain
- Informs capacity planning, health. Examples:
  - NXDOMAIN, DO bit %, v4/v6 query load
- Has good abstractions for extension
  - 1D-to-2D and 2D-to-<n>D data conversion
- Has good graphing for time series
  - Perhaps some JSON/XML download for offline processing would help with ‘what-if’ analysis
  - Constant y-axis for inter-graph comparison?

Daily traffic shapes (from DSC)



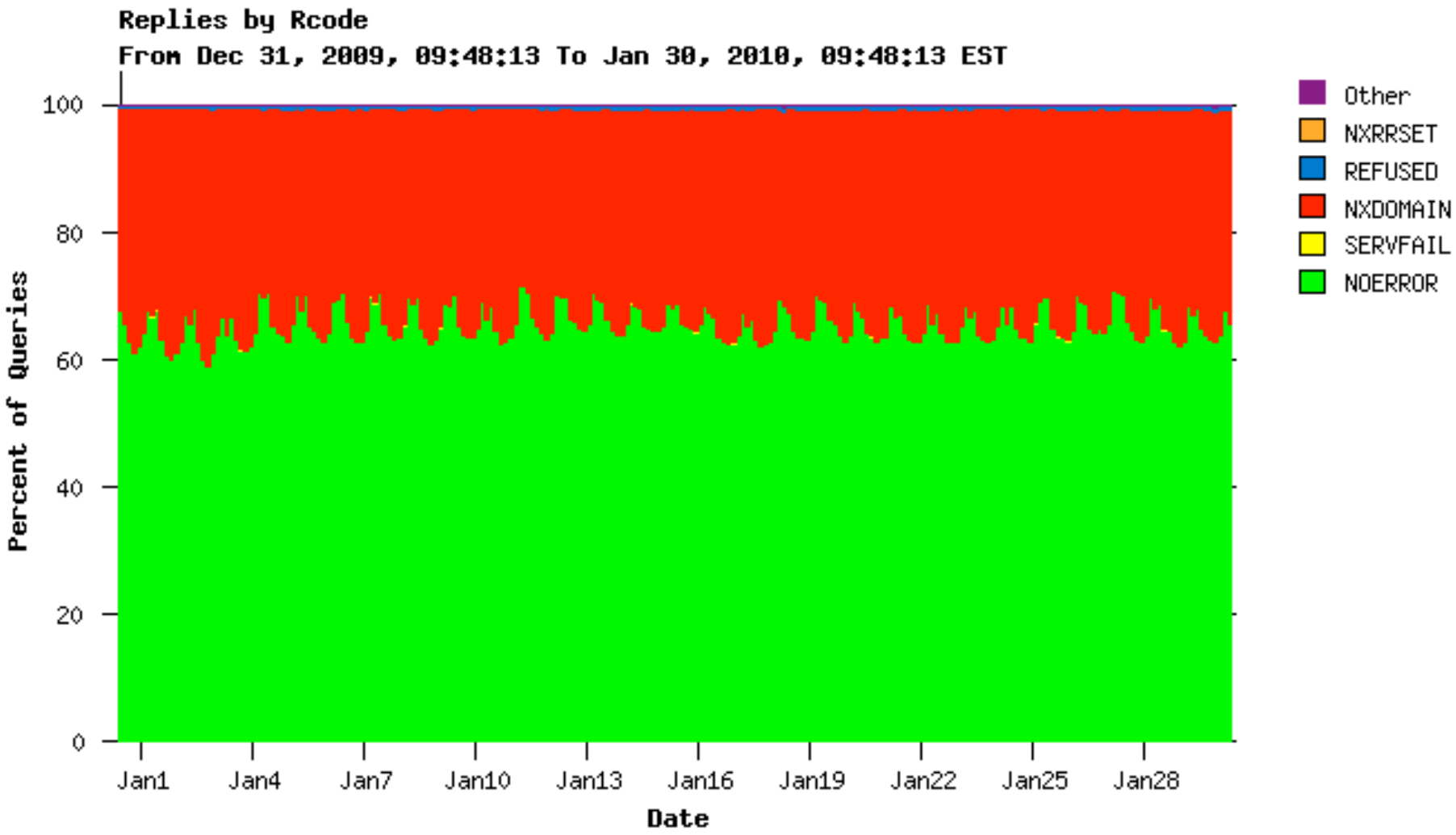
year-on-year trends (from dnscap)



Observations

- Allowing for time zone differences, dnscap/dsc seem to agree with each other (phew!)
- Strong single ‘artifact’ required some closer analysis (presented at ESNOG, RIPE)
  - JP ‘signal’ of DNS lookup
- Other ‘clock tick’ interval peaks visible
  - Consistent year on year.
  - Against background of constant(ish) load
  - Less ‘cron’ affected than we thought.
- Inter-site differences
  - Clearly volume, but also some of the artifacts
  - Strong unity of diurnal pattern for the NS (primary) server set, cross-site.

NXDOMAIN from DSC

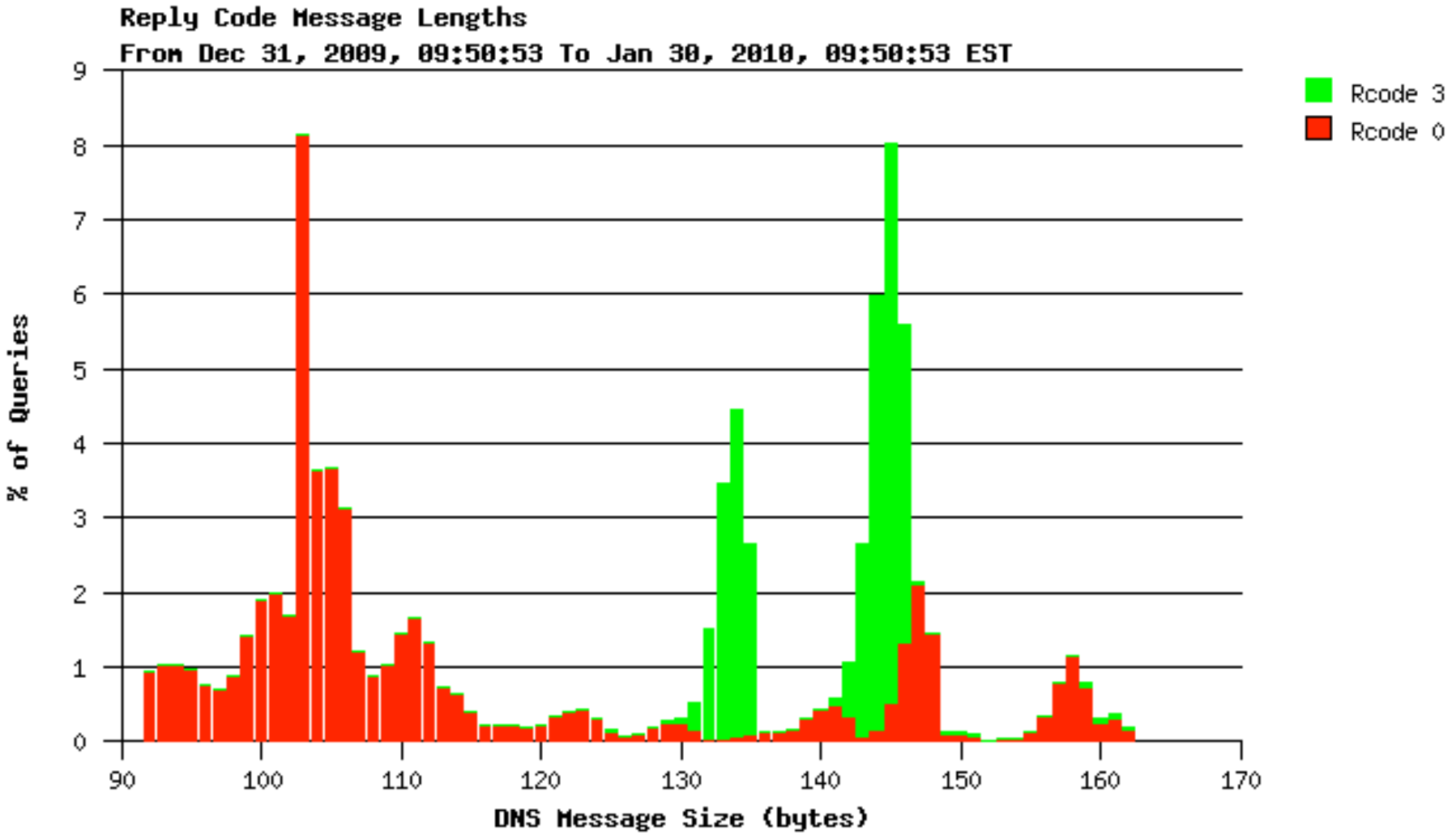


NXDOMAIN in reverse

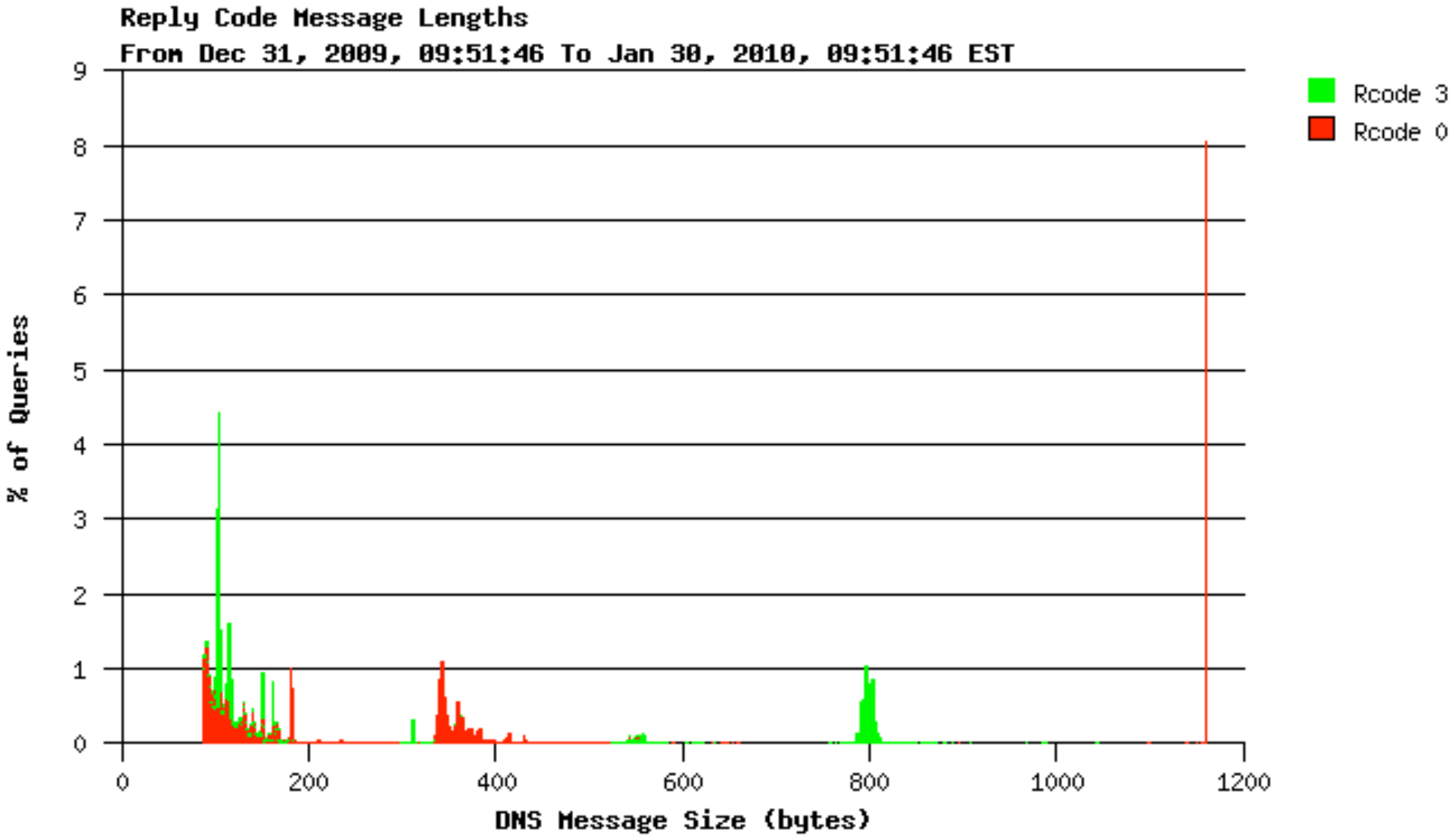
- NXDOMAIN of the order 40% of total PTR query load
  - Pre-DNSSEC, no significant cost difference to OK response
- Not a good sign of commitment to reverse-DNS.
  - Value proposition in reverse-DNS always ‘weak’
    - But, this can change. GeoIP, SRV, other uses can leverage
- Where is that diurnal sub-signal coming from?
  - Human-centric variances in applications use which causes broken reverse lookup?
    - (some of this is 256.255.abcdein-addr. nonsense)
- NXDOMAIN may be part of our measurement success
  - If its delegated, then the NS response probably caches and we’re not directly queried so much for that IP address reverse

What else is DSC telling us?

Response size/ NO DNSSEC (from DSC)



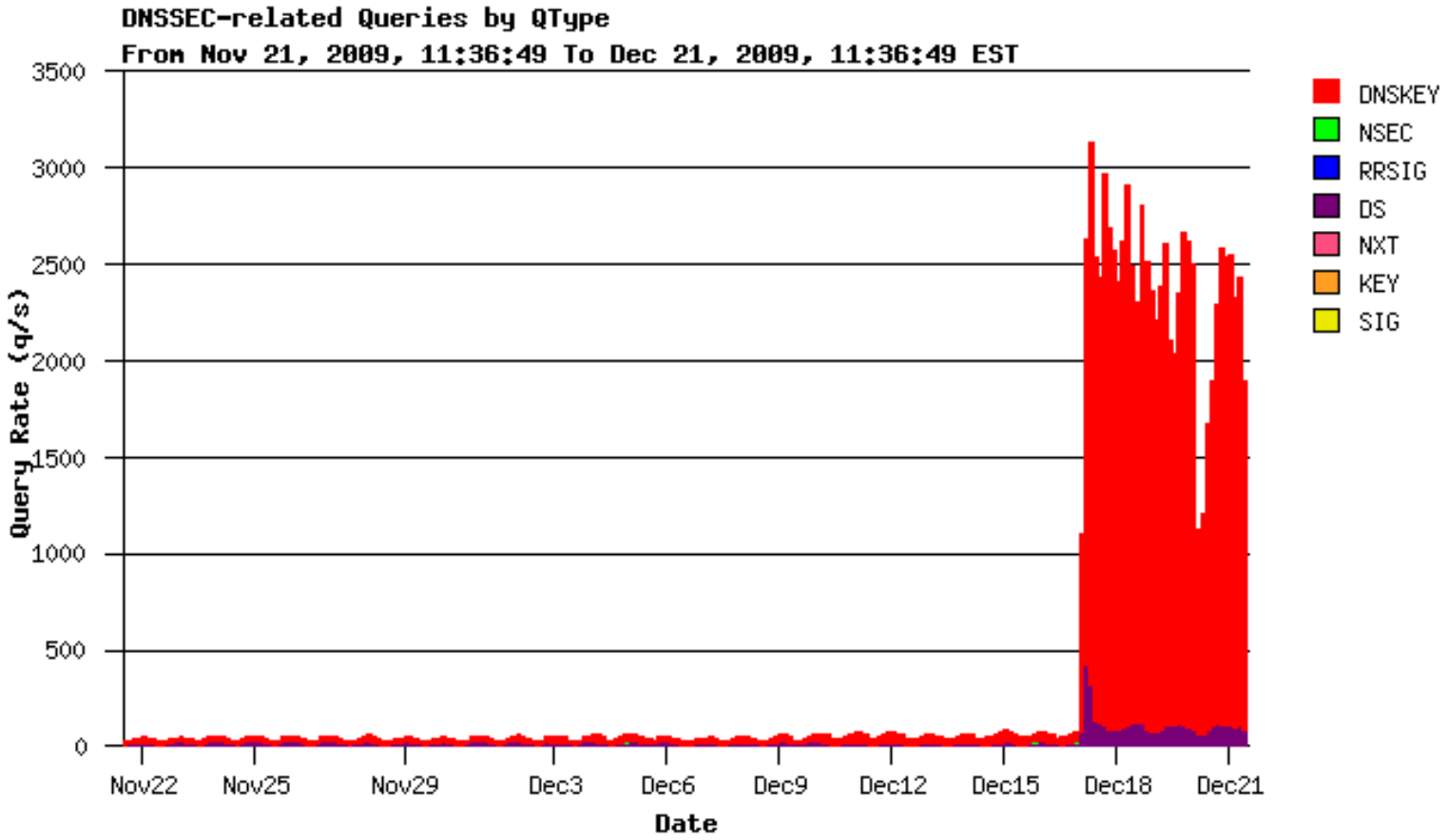
Response size/DNSSEC (from DSC)



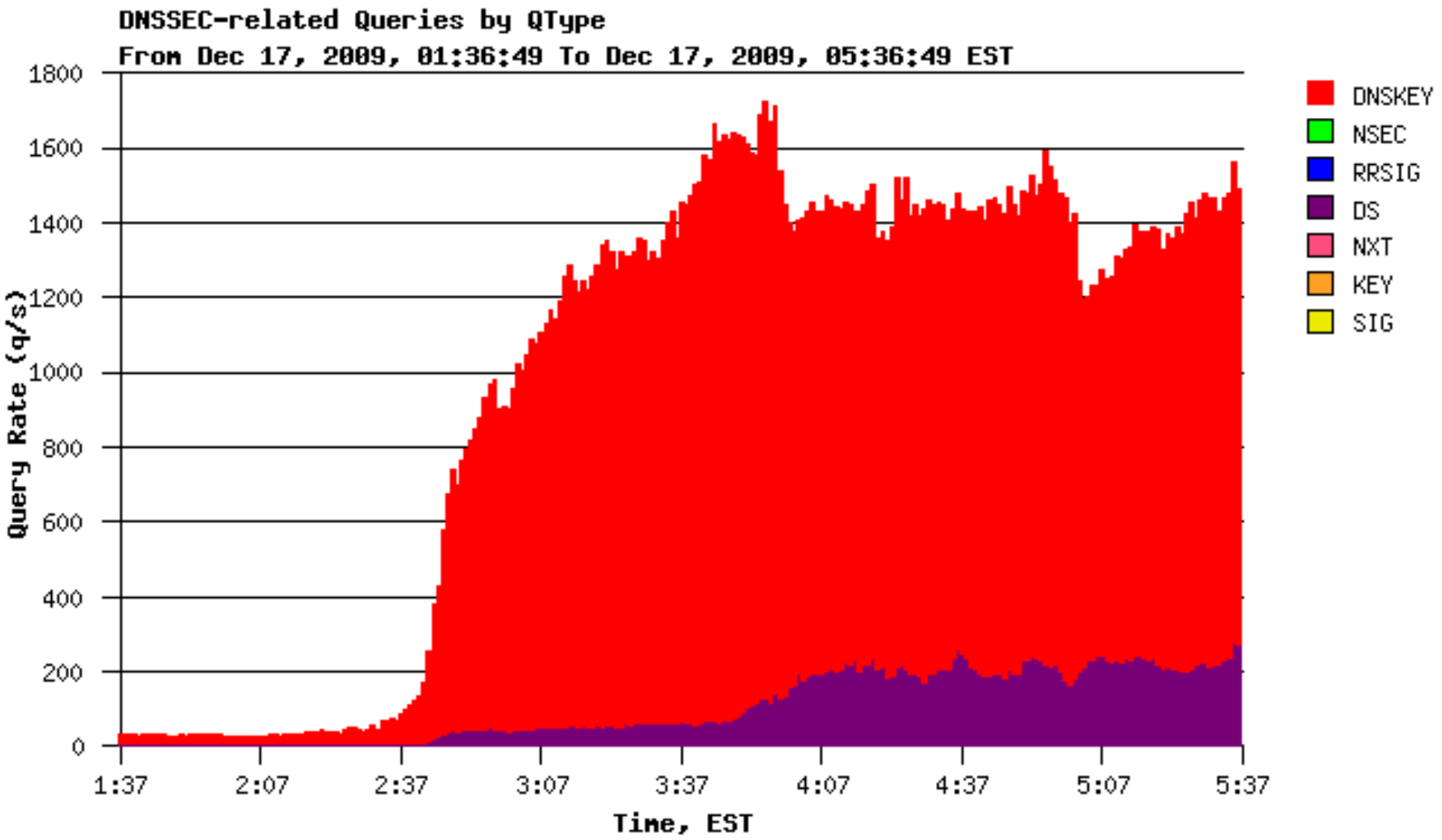
DNSSEC costs

- Pre-DNSSEC, response sizes close. (not identical)
- DNSSEC adds significantly to NXDOMAIN cost
  - RRSIG,sigset of NXDOMAIN including NSEC
  - 150byte response becomes 800 byte response
  - (in reverse, OK only jumps 100->400)
- Reverse has fixed-length(ish) queries and replies
  - So this response variance can be cleanly attributed to DNSSEC.
  - Outbound data cost increase, no inbound increase

Key rollover…



Key rollover (in detail)



DNSSEC validation costs

- RIPE still documenting issues, this is W-I-P
- Trust Roll causing problems
  - Clients don't update trust set
  - Mis-behaving clients not caching, backing off
    - 3000 q/s cycles to DNSSEC, DS of parent domains
- Remember: RIPE didn't do anything wrong, followed documented procedures, pre-announced the roll…
  - Root signing will help
  - Until the first key rolls? (KSK roll will expose?)
- New query load, large packets (sigsets)
  - Re-scaling required
  - No goodput improvement until deployment completed

Health/Holistic view of DNS

Health/Holistic view of DNS

- Most queries answered
  - In a timely manner
  - Correctly
- Service scaled to meet demand
- DNS as % of total traffic low
- Adapted over time
- Why are the queries being sent?
- The 40% NXDOMAIN…
- Obviously broken queries
- Impact of DNSSEC
- Resiliency
- DO bit variances

Goodput?

- I know the root ops get far far more of this…
- Significant volumes of dynamic dns update
  - Why do clients think they can do this?
- Significant volumes of 'mangled' queries
  - **`inet_pton()`** and **`inet_ntoa()`** not being used?
  - Hand crafted in-script DNS lookup of reverse?
- Mis-directed service location requests
  - Ubiquitous s/w (Microsoft, Apple) can do this promiscuously
- What *is* the ratio of DNS lookup to effective e2e?
  - Reasons to believe PTR has strong relationship to actual IP flows
    - (it mostly happens when a dst from {src,dst} receives IP)

Systems-wide redundancy

- Everyone wants secondary NS
  - Everyone winds up using the same secondary NS
- DNS services collapsing onto a small set of anycast providers
  - Leverage common infrastructure cost/efficiency
  - Works, but there are no magic bullets
- Resolver selection/fallback mismatch with client expectations
  - Browser driven demands for semi-immediate response
  - 30 second nserver timeout ubiquitous (untuned) too slow?
  - Parallel queries? First to reply wins?
    - Believe to drive IPv6 transport selection

Is RTT selection a myth?

- Much out-of-region DNS traffic
  - But evidence RTT of alternate NS much better
  - Therefore, why do they persist in coming to long/low RTT NS?
    - Because ‘good enough’ is coded in the library?
- Emergence of in-browser own-resolver code
  - Doesn’t follow the bind ‘norms’
  - What did RTT selection really mean anyway?
    - Demands patterns of DNS lookup, cache memory, persistence of data/state
    - Not applicable to the vast mass of 1-2 queries/day sources

What is reverse DNS telling us?

- Information from analysis of PTR query behaviors
  - Exposes {src,dst} relationships
    - Inter-provider relationships
    - Inter-economy relationships
- Rise of IPv6.
  - Tunneling exposed (6to4 address model)
    - 6to4 is 3x the size of the current native IPv6 market in NZ
  - Use of specific MAC (e164) exposed in ff:fe address plan
    - Eg 52% of all 6to4 in New Zealand is Apple
  - Only 43% of deployed IPv6 seen in 24h, against 89% of deployed Ipv4 in NZ
- Interesting, but doesn't inform on DNS health..
  - Except in the wider sense.. It does?

Things we don't do in our DNS monitoring

- We don't yet correlate the NS queries between our NS
  - What query source addresses are active at each node, and do they ask identical questions?
    - Would strongly suggest parallelism in the query.
    - A/AAAA questions might show hunting for transport.
    - Might show botnet or other storms on single queriers
- Requires disk/cputime investments
  - Or a protocol and some neutral combining logic

Things we don’t do (cont)

- We don’t do any ‘qualitative’ introspection
  - Zone convergence time as an ongoing report
  - RTT measures (eg leverage RIPE TTM)
- Follow up NXDOMAIN with resource holders
  - This could be part of a ‘full life cycle’ view of our DNS.
- We should do better DNS triggers in zenoss
  - Might have detected dnskey problem sooner

Things we might do differently

- Do in-server metrics do a better job of some of this?
  - Not yet a common reporting platform?
- Costing measurement/reporting up-front
  - Ongoing capex (disk space mostly)
- Uplifting data to 'home' getting harder
  - Move to the cloud?
  - (easier to do inter-site comparisons if data in one place)

Clue Density Dropping

Additional clue sought!