What part of "NO" is so hard for the DNS to understand?

Geoff Huston

Joao Damas

APNIC Labs

We were looking elsewhere…

- We were setting up a measurement experiment that was looking at the extent of support for aggressive NSEC caching (RFC8198) in the DNS
- The experiment setup involved presenting to the user a DNS name that did not exist from a signed zone, so that we would pass an NSEC record to a DNSSEC-aware resolver
- But was was intriguing was that we were seeing many more queries for the non-existent name than we had expected

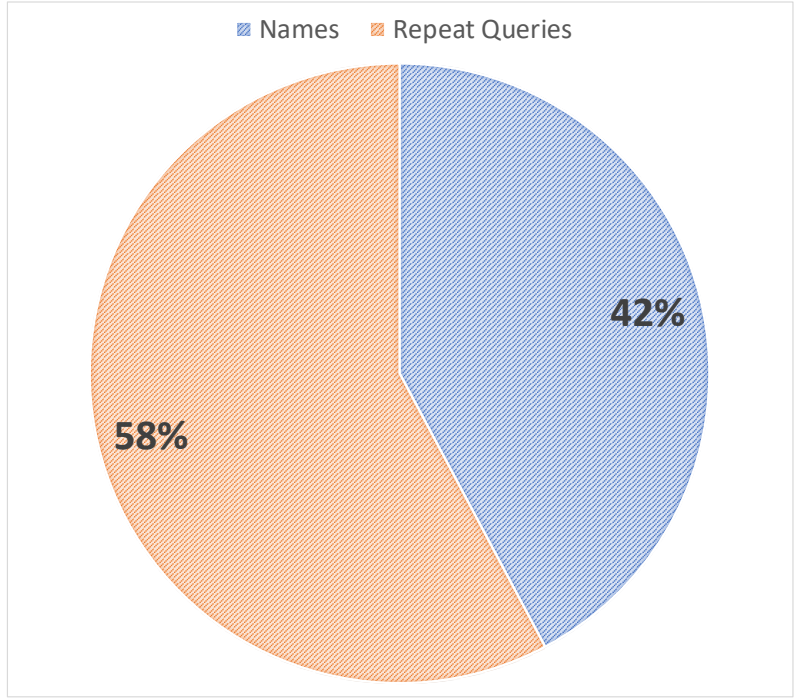
What we saw:

- We used an online ad to get users to query for a unique non-existent DNS name
- And then counted the number of queries we saw for these names

Queried Names: 60,210,983

DNS Queries: 142,631,272

- That's an average of 2.37 queries per non-existent name!



Why so many queries?

Expectations

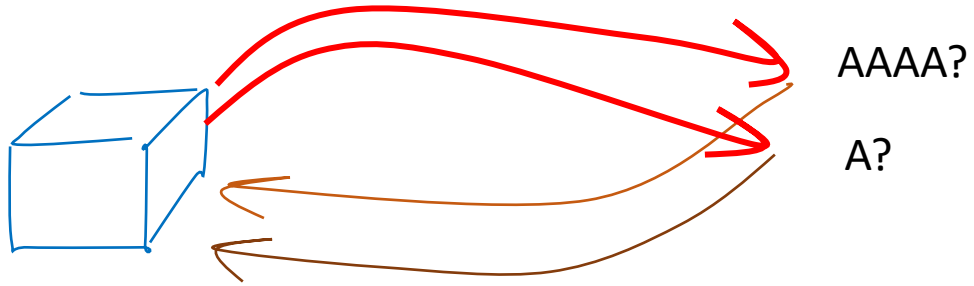
- If “NO means NO” then naïvely we would expect to see 1 query per name, not 2.37 queries per name

- But maybe that’s just too naïve these days…

Happy Eyeballs and the DNS

A ‘happy eyeballs” dual stack client will launch 2 DNS queries back-to-back (roughly), for A and AAAA records of the name

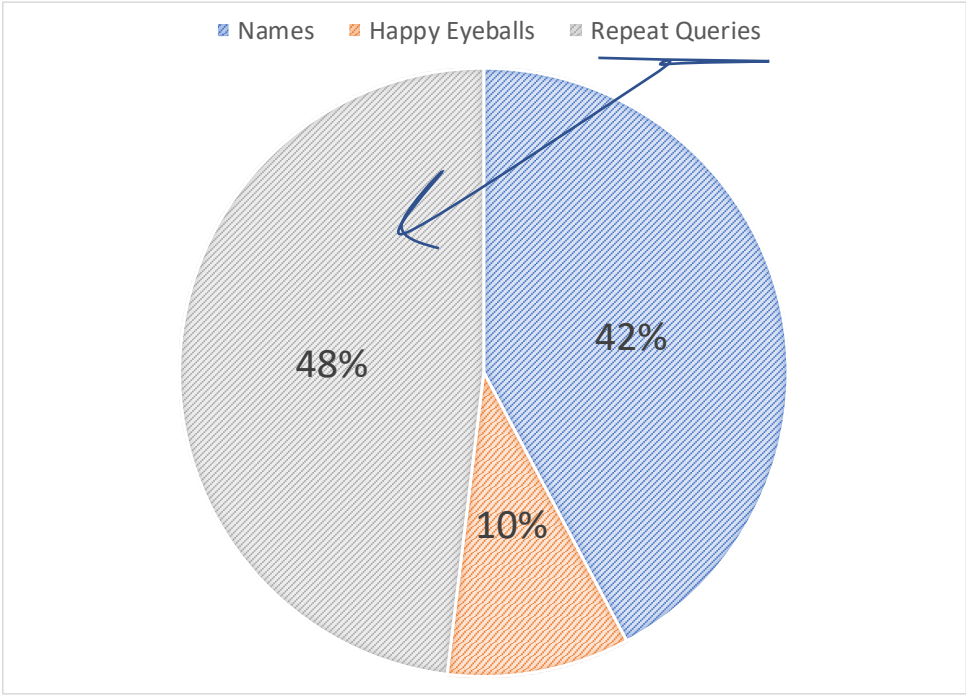
- 23% of clients asked both A and AAAA records
- 3% asked only AAAA records (*)
- 74% asked only for A records



* If the client asks for a AAAA record and waits for a response before asking for the A record then the NXDOMAIN response will stop the connection process and any subsequent A query will not be performed

Factoring Happy Eyeballs

- If we split out the A and AAAA queries the experiment launched 73,537,852 DNS resolution ‘events’
- We saw 142,631,272 DNS queries, or an average of 1.93 queries per name
- That’s better, but still unexpectedly high



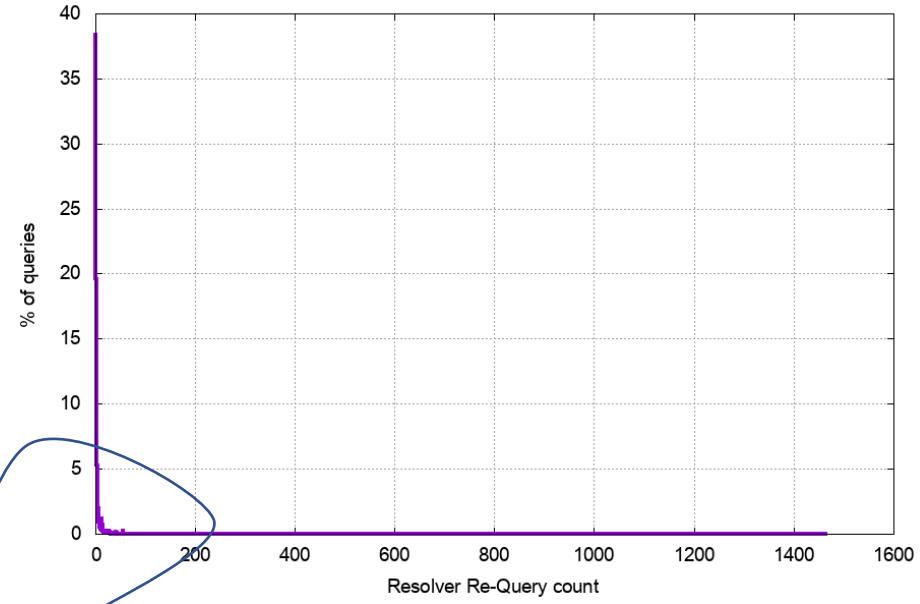
Single vs Multiple Queries

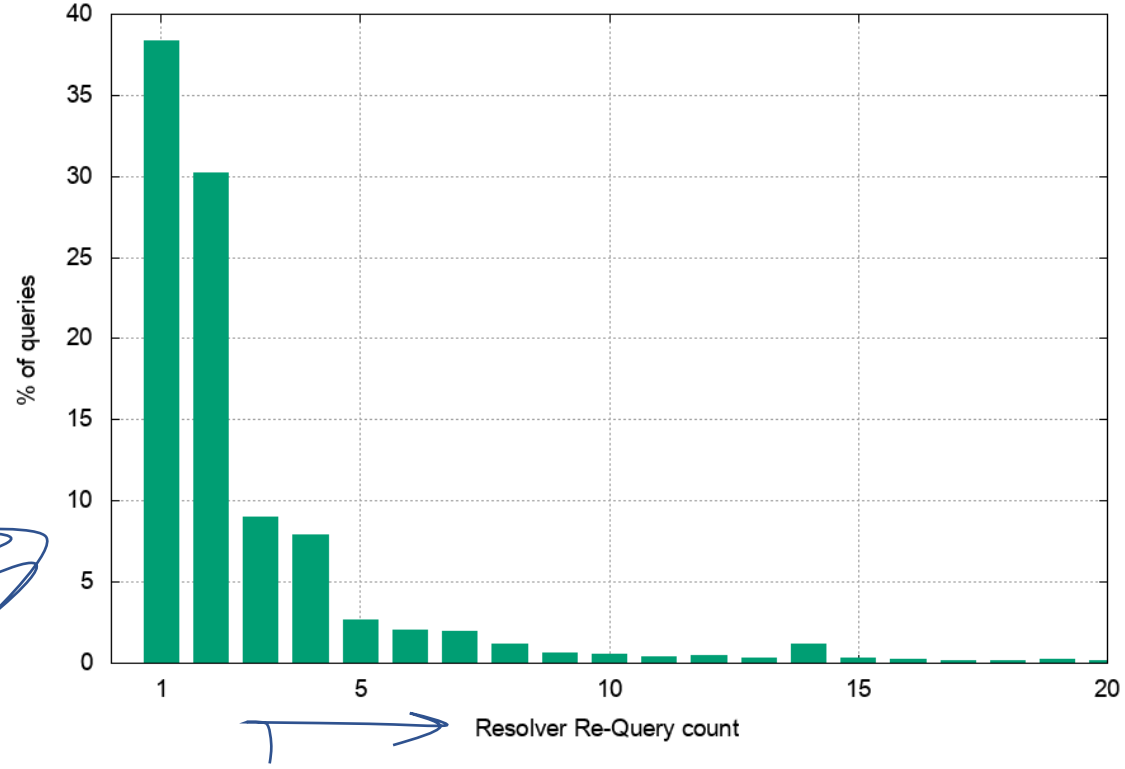
- 36,059,484 resolution ‘events’ were completed with just 1 query to the authoritative server (49% of all resolution events)
- If there were multiple queries for a name (>= 2 queries(, then the average of the multiple queries was 2.84 queries

Distribution of Queries

- Is this a generic issue of re-queries across a large set of queries?
- Or a small number of queries that are the subject of a frenzy of re-queries?

Re-query Distribution



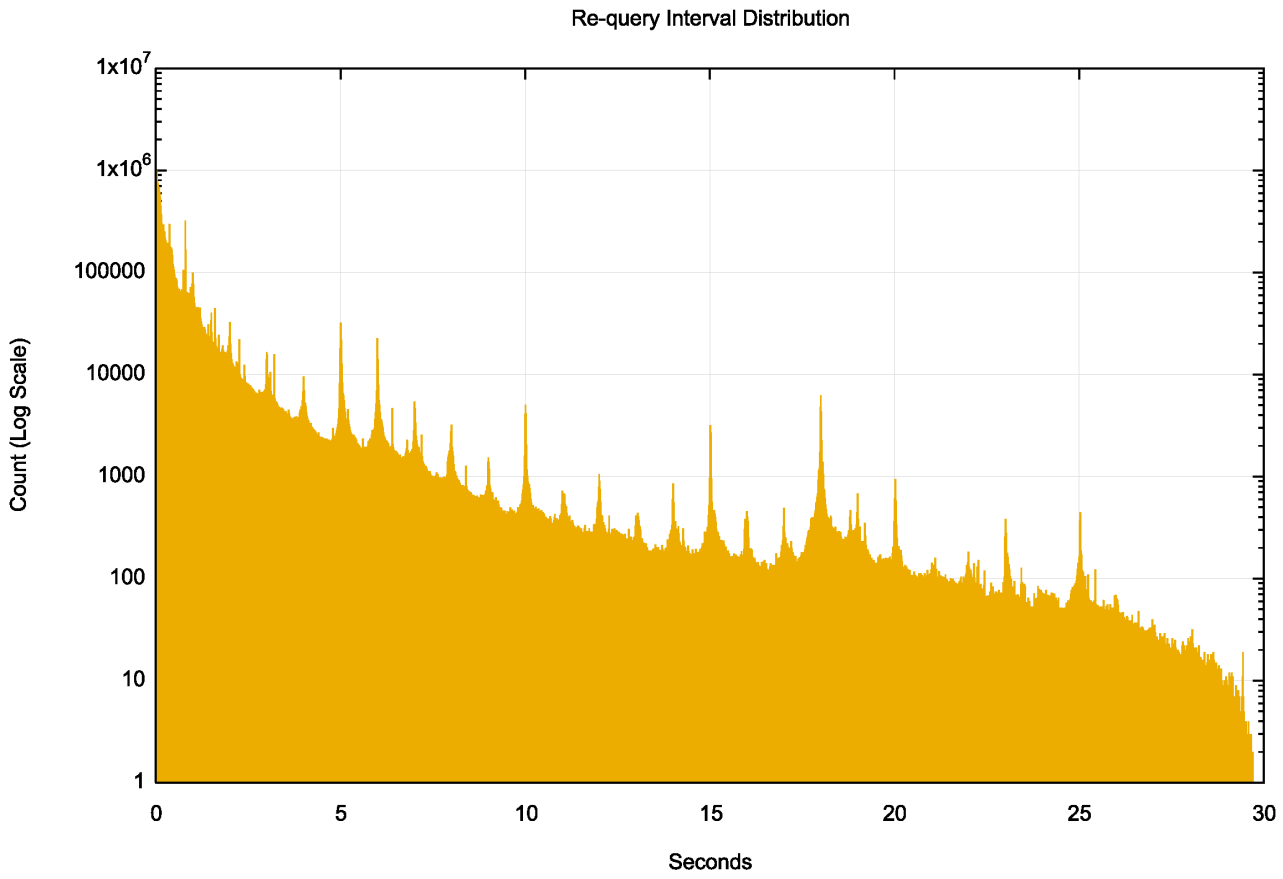


32% of queries were parts of query sequences of 3 or more

Does UDP suck THAT much?

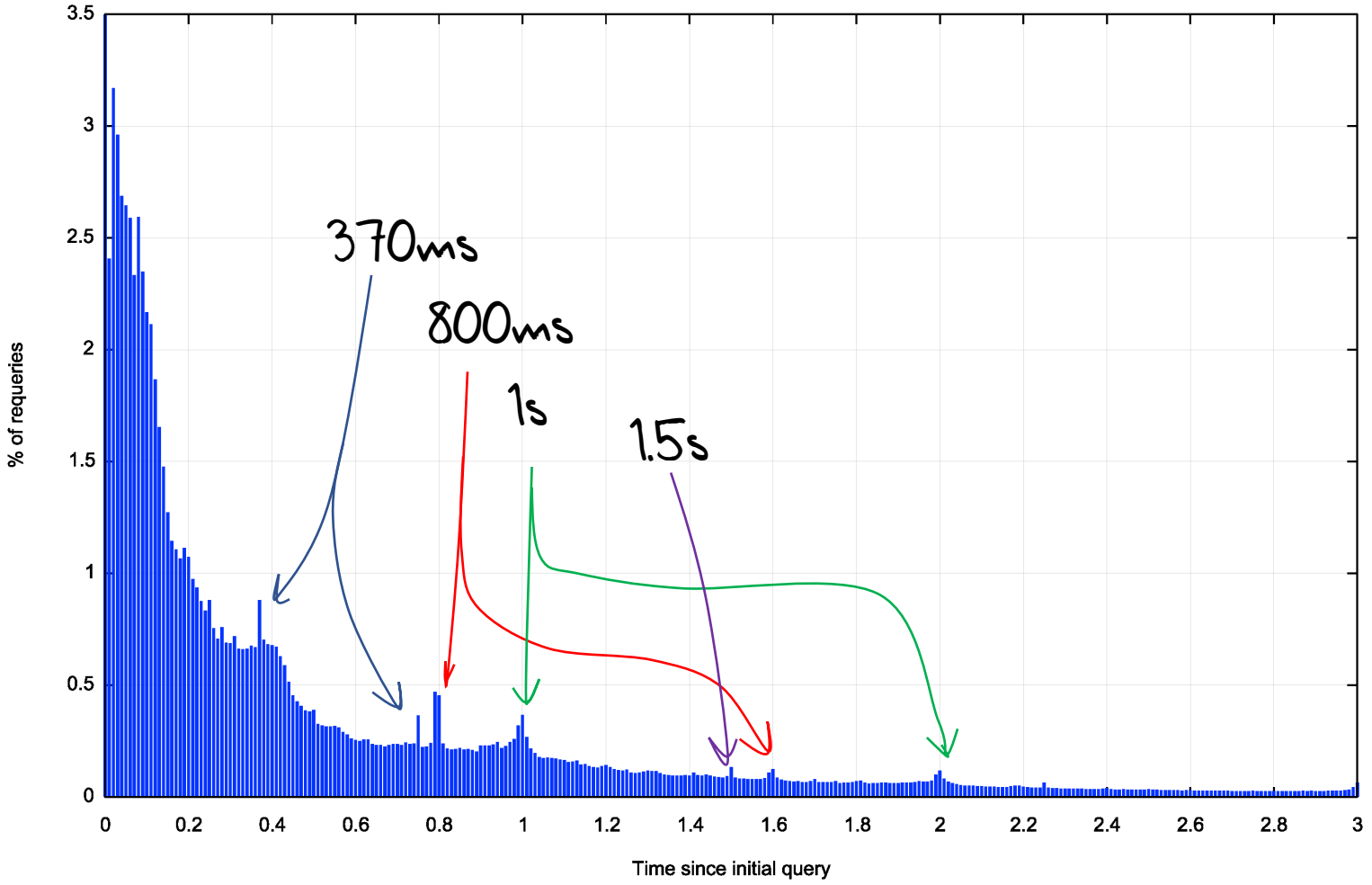
- Why is the total re-query rate at 51% of tests?
- Surely DNS over UDP is not THAT bad
  - The servers are responding to every query
  - The signed response is 603 bytes in size
  - We are using a distributed setup of servers to localize DNS transactions
  - So why are the servers seeing 51% of tests generate 2 or more queries?

Re-Query Time Intervals



There are strong local peaks at regular 1 second intervals – this would appear to be an end host re-query behaviour

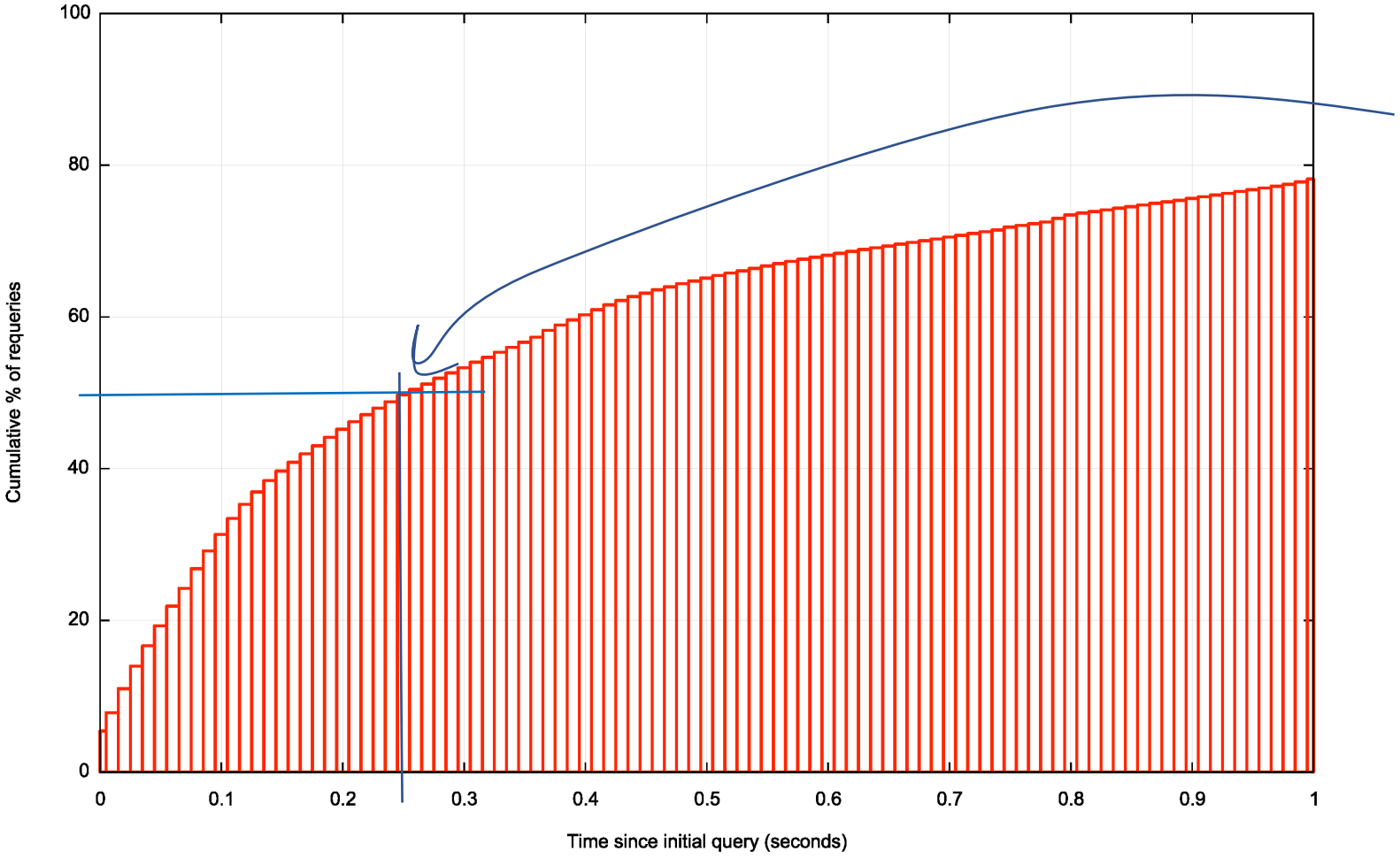
Re-Query Time Intervals



There are local re-query peaks here at 370 ms, 800ms, 1 sec and 1.5sec

It is likely that these time intervals represent recursive resolver re-query timers

Re-Query Time Intervals



One half of all re-queries occur in the first 250 ms

DNSSEC?

- This is a DNSSEC-signed non-existent name.
- Is DNSSEC a factor in the excessive re-query volume?
  - i.e. is the additional time to validate causing requery timers to trigger?
- We added an unsigned non-existent name to the test set

Signed vs Unsigned

	Signed	Unsigned
Experiments	65,686,452	69,251,349
A/AAAA	81,057,694	84,979,990
Queries	153,697,947	122,665,888
Single Query Exps	47,694,930	60,061,746
Ratio	59%	71%
Multi-Query Exps	33,092,764	24,918,244
Re-Query Rate	3.19	2.51

Split out the 'happy eyeballs' factor

DNSSEC validation adds delay, and in around 12% of cases this additional delay causes the resolver system to re-query the name

Signed vs Unsigned

- So there are a set of resolvers that are working on the margins of their timers, and in 12% of cases the additional DNSSEC queries to perform validation cause the resolver to time out and re-query.

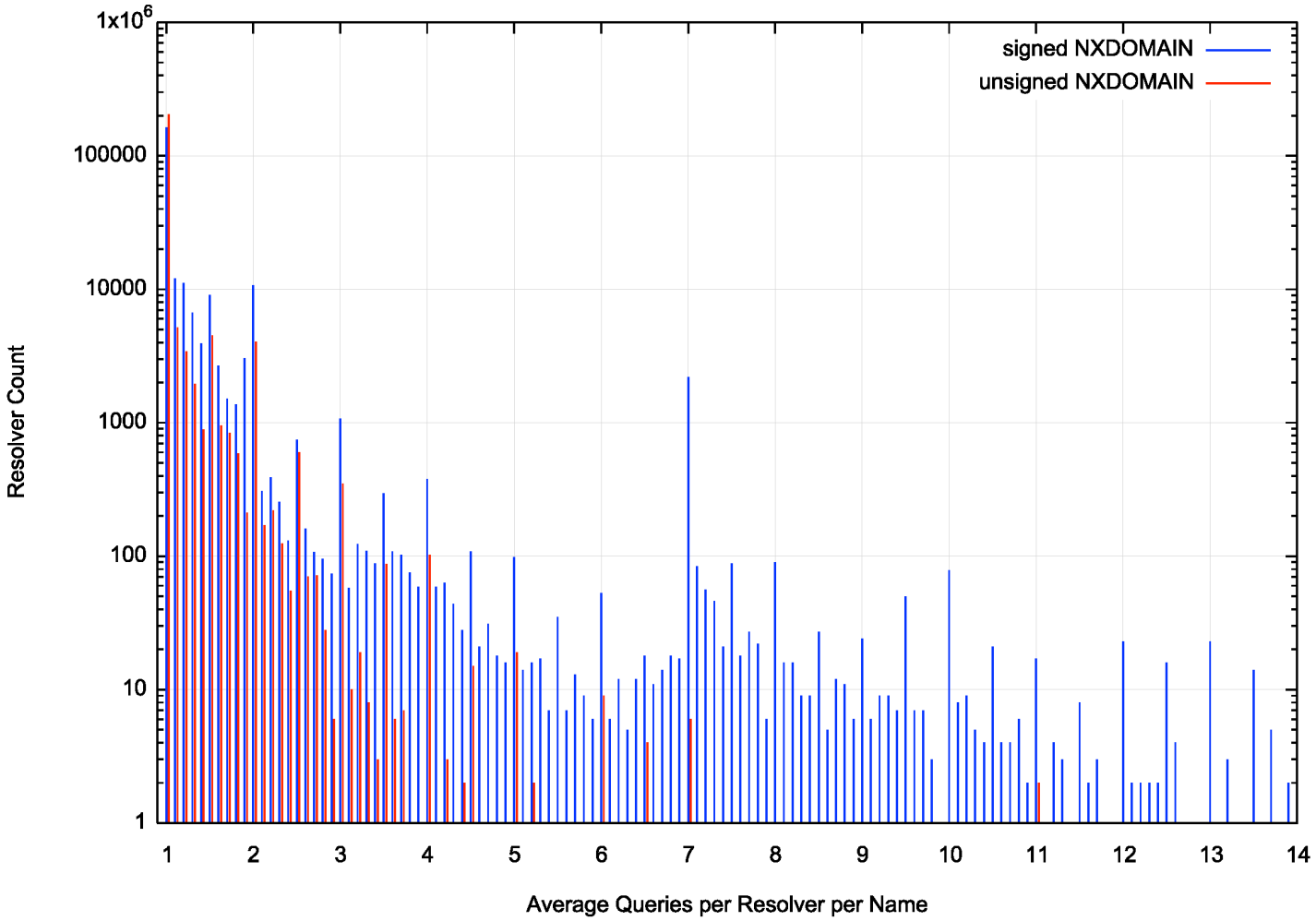
- OK – but the 12% delay factor is still not enough to explain the high query rate for NXDOMAIN responses
- What else is going on?

NXDOMAIN re-queries

- 39% of queries are re-queries when the domain is signed
  - And of those, 30% are the same source address and 69% are the same subnet
- 29% of queries are re-queries when the domain is unsigned
  - And of those, 11% are the same source address and 50% are the same subnet

- It appears that DNSSEC validation adds a time component that causes resolvers to trigger internal timers and re-query in around 12% of cases

Re-Queries per resolver IP addresss



The log scale exaggerates the effect, but we observe that a DNSSEC-signed NXDOMAIN response generates a higher repeat query profile from the same resolver IP address

Resolver "farms"

We also see query patterns of the form:

Resolver	Query Time
7x.xxx.0.178	0.752
6z.zzz.161.146	0.865
7x.xxx.0.230	0.980
6z.zzz.161.220	1.094
7x.xxx.0.188	1.201
6z.zzz.161.182	1.319
7x.xxx.0.180	1.430
6z.zzz.161.144	1.542
7x.xxx.0.226	1.650
7x.xxx.0.138	1.654
6z.zzz.161.134	1.762
6z.zzz.161.222	1.775

It appears that some resolver farms operate by farming the query across all members of the farm. This pattern seen here shows two such cases where different IP addresses in the same subnet repeat the initial query at approximately 100ms intervals

How common is this form of subnet-based query repetition?

Re-query Profile

	Signed	Unsigned
Queries	153,697,947	122,665,888
Re-queries	59,782,873	35,297,618
Same IP Address	17,848,729	4,024,583
Same Subnet	41,111,416	17,618,192

Some 70÷ of the re-queries are from resolvers that share the same subnet prefix when the domain is DNSSEC signed. This drops to 50÷ with an unsigned domain

This points to some outstanding issues with resolver farm management

It may sound odd but…

- Is NXDOMAIN part of the issue here?
- Is the re-query rate lower if the name exists in the DNS?
- So we also ran the same query count for queries to a dual-stack defined domain name that did exist in the DNS

NXDOMAIN vs A/AAAA re-queries

NXDOMAIN Signed:

41% of experiments generate multiple queries

39% of queries are re-queries (avg of 3.19 queries per experiment)

A/AAAA Signed:

18% of experiments generate multiple queries

13% of queries are re-queries (avg of 5.81 queries per experiment)

NXDOMAIN vs A/AAAA re-queries

NXDOMAIN Signed:

41% of experiments generate multiple queries

39% of queries are re-queries (avg of 3.19 queries per experiment)

A/AAAA Signed:

18% of experiments generate multiple queries

13% of queries are re-queries (avg of 5.81 queries per experiment)

NXDOMAIN Unsigned:

39% of experiments generate multiple queries

29% of queries are re-queries (avg of 2.51 queries per experiment)

A/AAAA Unsigned

38% of experiments generate multiple queries

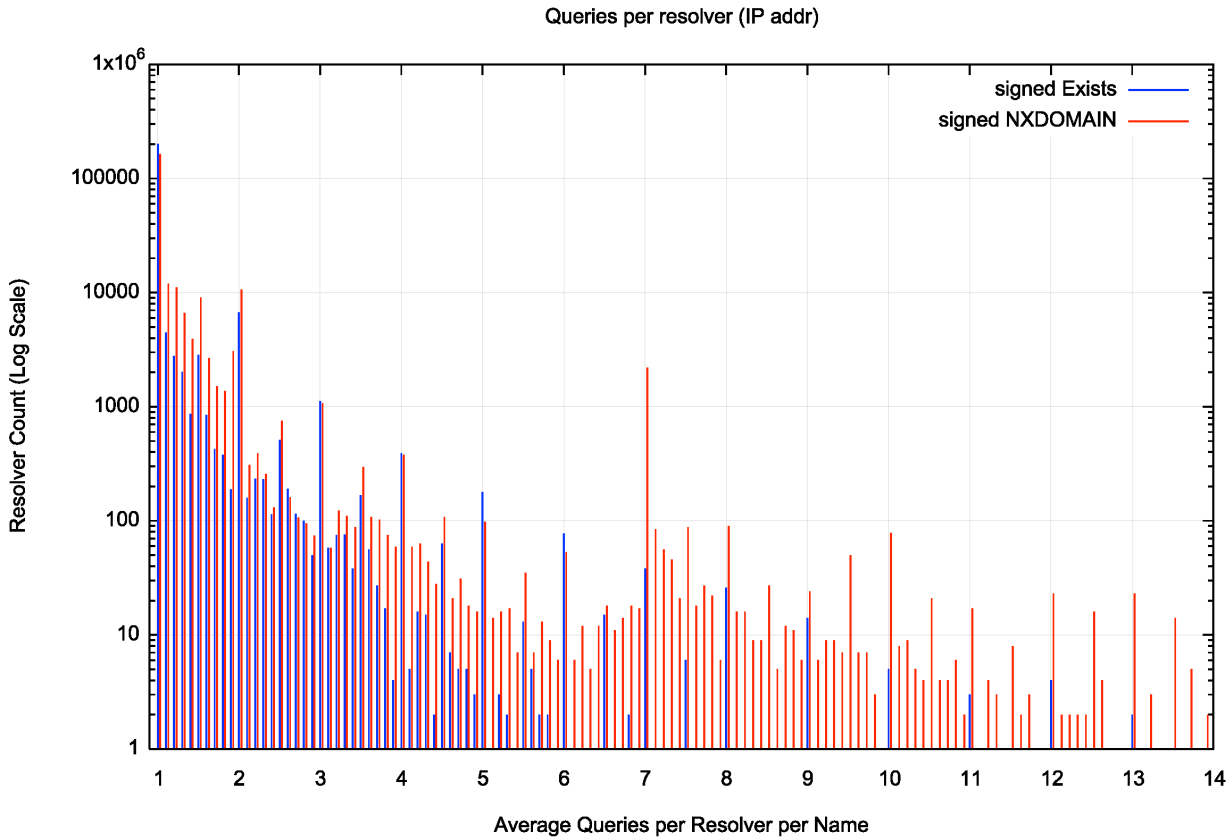
36% of queries are re-queries (avg of 3.03 queries per experiment)

How Odd!

- The proportion of experiments that are completed in a single query are unchanged with there is an unsigned response
  - Where re-query occurs the average number of queries per experiment rises from 2.5 to 3.0
- The proportion of experiments that are completed in a single query rise with there is a signed response
  - Where re-query occurs the average number of queries per experiment rises from 3.2 to 5.8

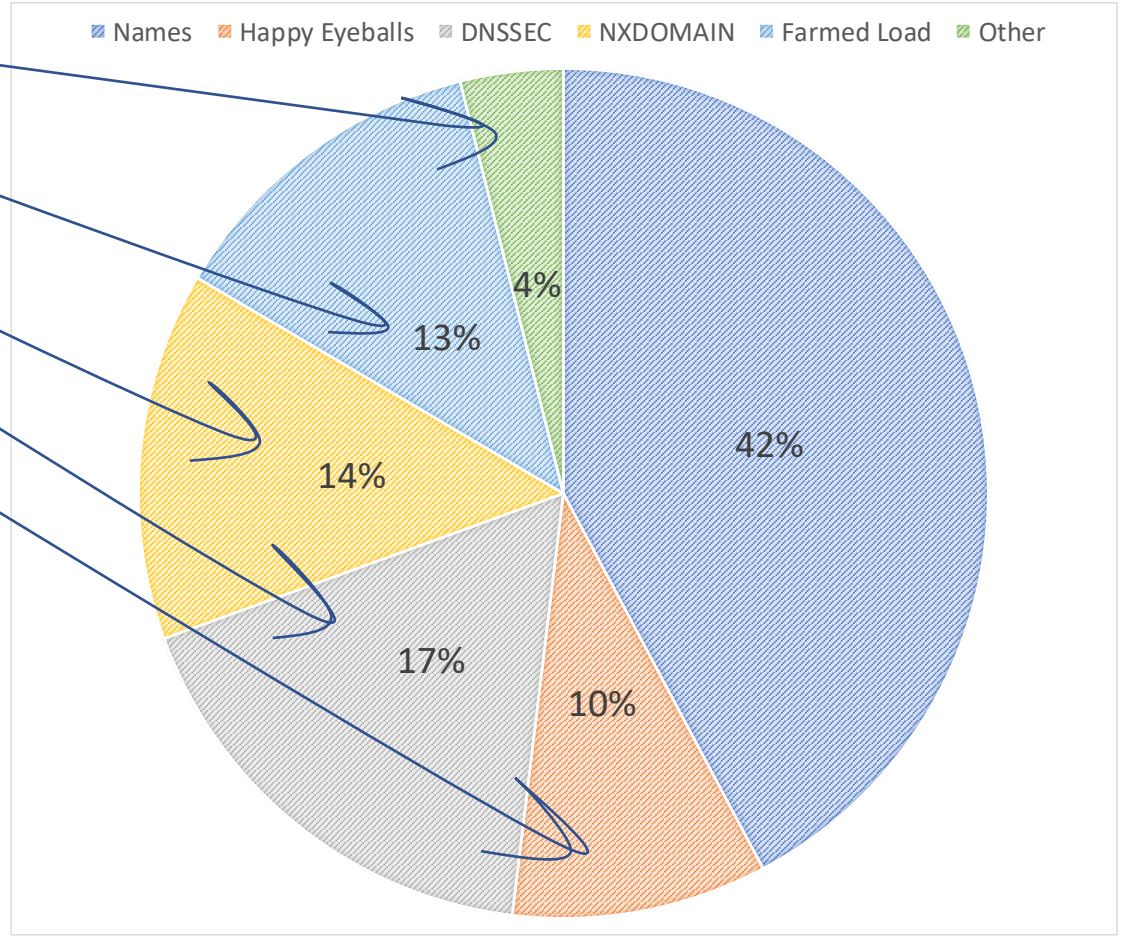
WTF?

- A DNSSEC-signed NXDOMAIN response generates more re-queries than a DNSSEC-signed A / AAAA response



Pulling it back together

- Why are there so many repeat queries?
  - <reasons>
  - Resolver Farms
  - NXDOMAIN
  - DNSSEC Signing
  - Happy Eyeballs



Thanks!