DNS Resolution Behaviour (DNS Measurement)

Geoff Huston & Joao Damas
APNIC Labs

Back in 2019, 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 what was intriguing was that we were seeing many more queries for the non-existent name than we had expected!

What we saw in 2019:

- 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!



Expectations

- If “NO means NO” then we would expect to see 1 query per name, not 2.37 queries per name resolution task
  - But maybe that’s just too naïve an expectation!

- After all - the DNS is now SO clever that this just couldn’t be due to random DNS noise – could it?

We returned to this experiment in 2026

Data gathering was performed over 7 days – 5th August through to the 11th August 2026, using a server that always returned NXDOMAIN to all query names, and we used the Ad-based measurement system to get a broad sample of users to resolve a unique name from this server

Over this period, we ran:

- **115,799,726** NXDOMAIN tests,

and recorded

- **512,740,759** queries

That's an average of **4.43** queries per name – up from the average value query multiplier of **2.37** that we observed in 2019

That's a significant jump. What's going on?

In the DNS, QDCOUNT is 1!

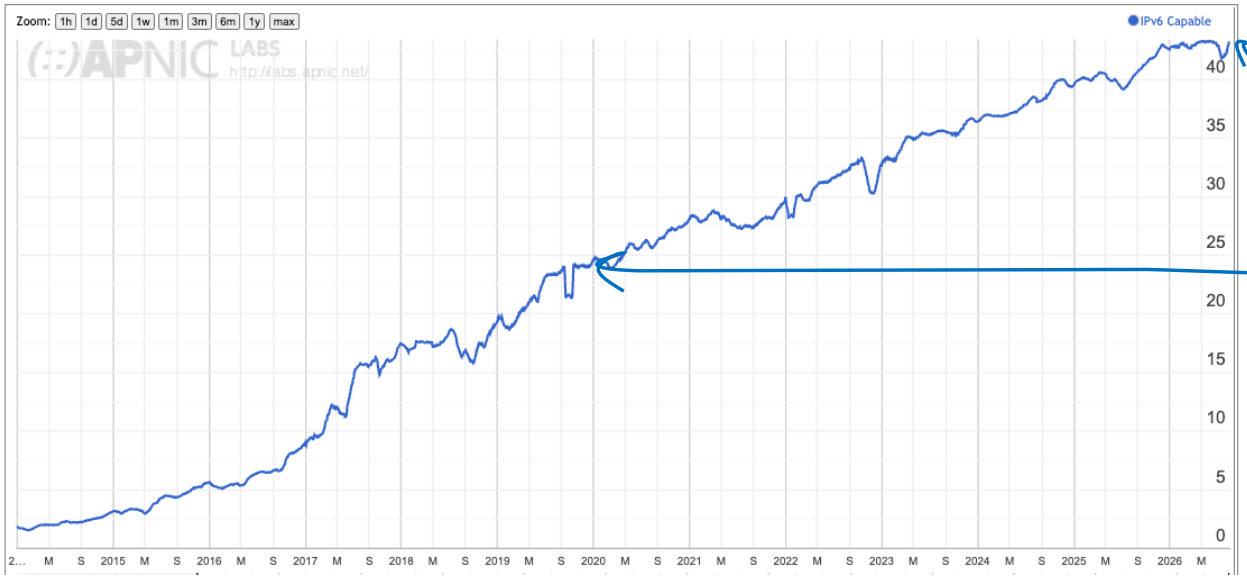
- RFC 9619 formalizes common practice where in a standard DNS query the count of queries (QDCOUNT) is limited to 1
- What if a client is both IPv6-capable and QUIC-capable and wants to know the application server's capabilities?
  - The client will need to separately launch 3 DNS queries with Query Types of A, AAAA and HTTPS
  - It is impractically slow to do this in a serial manner (waiting for a response before launching the next query), and because the client can't pack all three queries into a single DNS transaction, then the client will fire off three queries in rapid succession
- How does this observation relate to query repetition increases since 2019?

What's changed since 2019?

Two major changes are:

1. The increase in the deployment of IPv6 => rise in AAAA queries

Use of IPv6 for World (XA)



August 2026 – 43% of users

October 2019 – 24% of users

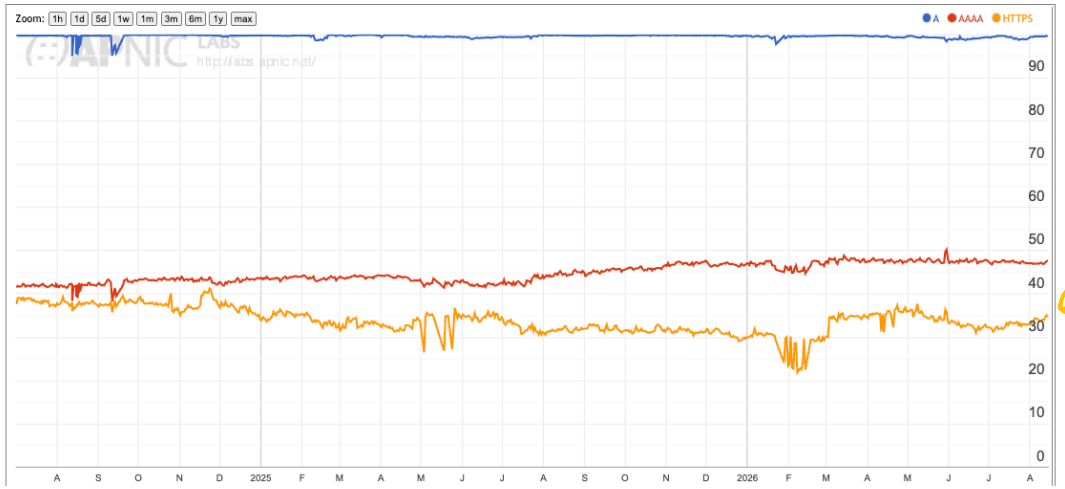
https://stats.labs.apnic.net/ipv6/XA?c=XA&x=1&v=1&p=1&r=1&w=30&p=0

What's changed since 2019?

Two major changes are:

1. The increase in the deployment of IPv6
2. The introduction of the HTTPS query

Use of HTTPS Queries in World (XA)



August 2026 – 32% of users

October 2019 – 0% of users

https://stats.labs.apnic.net/qhttps/XA

Distribution of Query Types

- A ‘happy (web) eyeballs” dual stack client will launch 2 DNS queries back-to-back (roughly), for A and AAAA records of the name
  - 49% of endpoints queried for both A and AAAA records (up from 23% in 2019)
  - 50% of endpoints queried for A records only (down from 74%)
  - 1% of endpoints queried for AAAA record only
- An HTTPS-aware client will also launch an HTTPS DNS query
  - 32% of endpoints queries for HTTPS records
- With our 115M tests, this distribution of client capabilities would result in 217M queries if each of these query types were queried just once

Distribution of Query Types

- A ‘happy (web) eyeballs” dual stack client will launch 2 DNS queries back-to-back (roughly), for A and AAAA records of the name
  - 49% of endpoints queried for both A and AAAA records (up from 23% in 2019)
  - 50% of endpoints queried for A records only (down from 74%)
  - 1% of endpoints queried for AAAA record only
- An HTTPS-aware client will also launch an HTTPS DNS query
  - 32% of endpoints queries for HTTPS records
- With our 115M tests, this distribution of client capabilities would result in 217M queries if each of these query types were queried just once

But we observed a total of 512M queries.

- The remainder of the observed query set (~300M queries, or 57%) is due to some form of query repetition

What about Happy Eyeballs for the DNS?

Are resolvers performing “happy eyeballs” in DNS resolution?

- Here “happy eyeballs” means launching simultaneous DNS queries using IPv4 and IPv6 transport
- We are used a 30ms timer as a measure of “simultaneous” queries, and we are looking for the number of tests where queries associated with each test use “simultaneous" queries over IPv4 and IPv6 transport
- We observed some 35M cases where there were 2 or more queries in the first 30ms, and of these some 2.4M cases (7%) had queries over IPv4 and IPv6 transport.
- This reduces the overall DNS repetition set from 300M to 297M
  - i.e. “Happy Eyeballs” does not appear to be a significant factor behind query repetition

What about Happy Eyeballs for the DNS?

Are resolvers performing “happy eyeballs” in DNS resolution?

- Here “happy eyeballs” means launching simultaneous queries using IPv4 and IPv6 transport
- We are used a 30ms timer for “simultaneous” queries, and we are looking for clients where queries associated with "simultaneous" queries over IPv4 and IPv6 transport
  - We observed some 35M cases where there were 2 or more queries in the first 30ms, and of these some 2.4M cases (7%) had queries over IPv4 and IPv6 transport.
- This reduces the overall DNS repetition set from 300M to 297M
  - i.e. “Happy Eyeballs” does not appear to be a significant factor behind query repetition

NO – the DNS does not do that!

UDP?

- A total query repetition rate of 57% of queries is extremely high!
- Is this due to UDP timeouts?

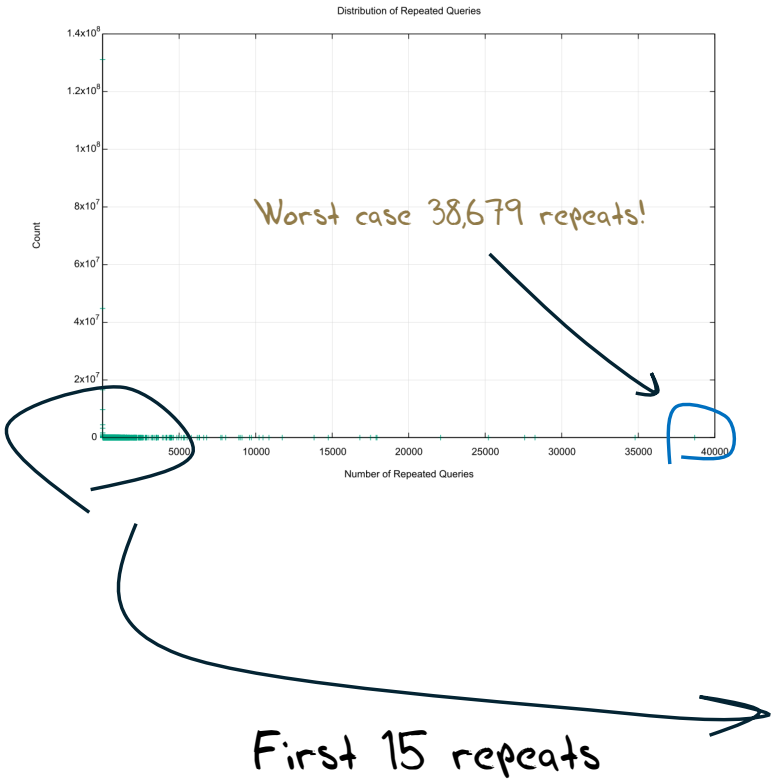
UDP?

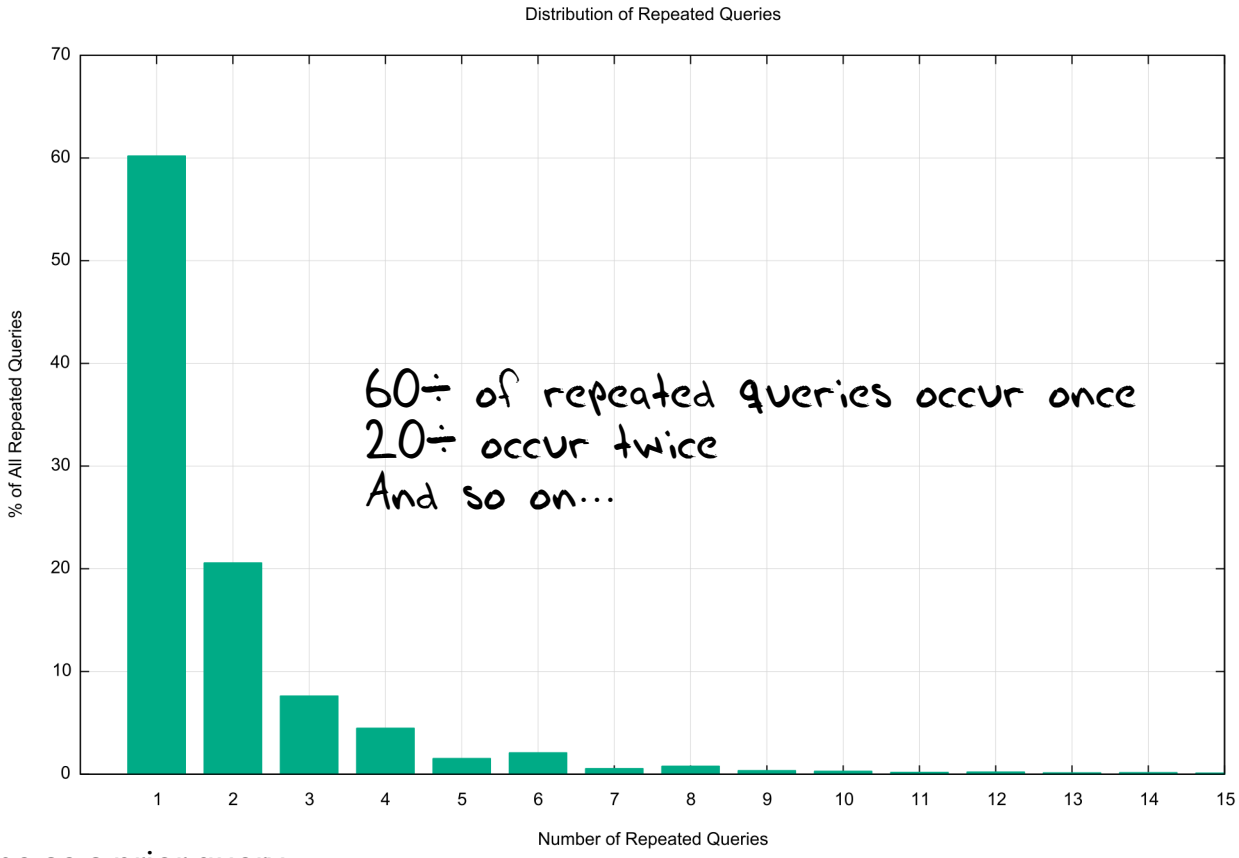
- A total query repetition rate of 57% of queries is extremely high!
- Is this due to UDP timeouts?
- Surely UDP is not THAT bad!
  - The servers are responding to every query
  - The response is 603 bytes in size
  - We are using a distributed setup of servers to localize DNS transactions
  - It's highly unlikely that we are experiencing a high UDP response drop rate
  - So why are our servers seeing 57% query repetition, and 41% of tests generating repeated queries?

Why would a DNS query be repeated?

- DNS relies on UDP transport by default
- As a result, the querier receives no notification when a query or response is lost
- When UDP responses are discarded, the server consequently sees a high rate of repeated queries
- To infer packet loss, the querier uses a timer and resubmits the query if no response arrives before it expires
  - An excessively long timer can delay DNS resolution
  - Shorter timer settings (below the actual response time) can improve DNS responsiveness when UDP packets are lost, but they also cause unnecessary query repetition in other circumstances
- Where some UDP packet loss exists, aggressive UDP timer settings would therefore be expected to improve overall DNS responsiveness.

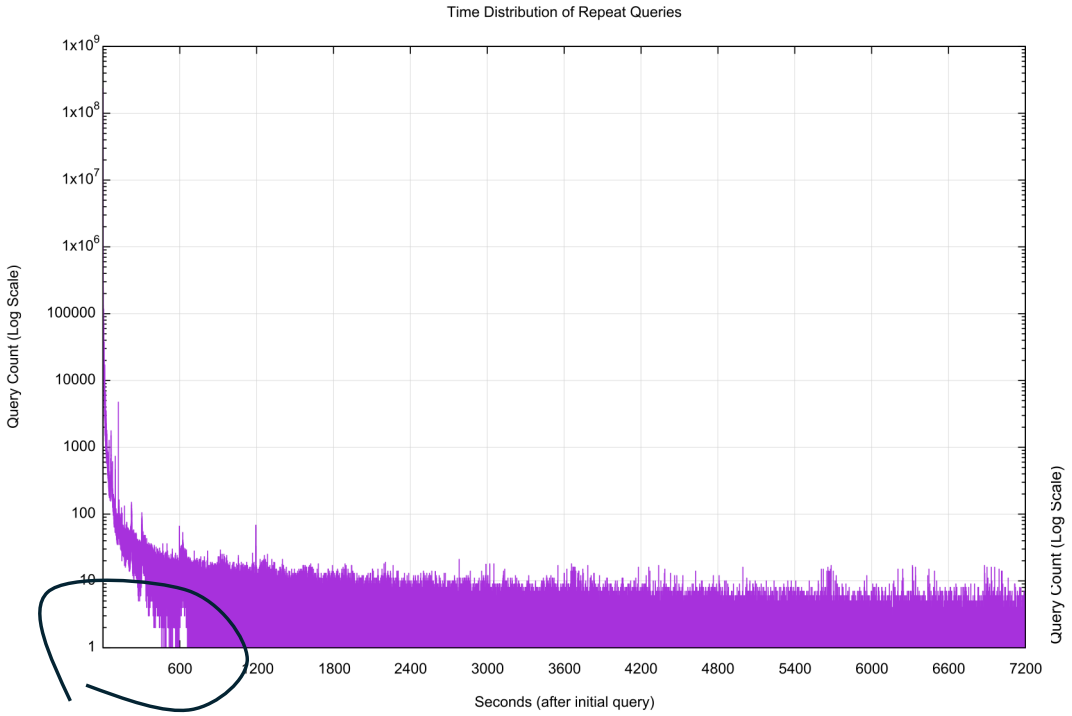
Distribution of Repeated Queries*

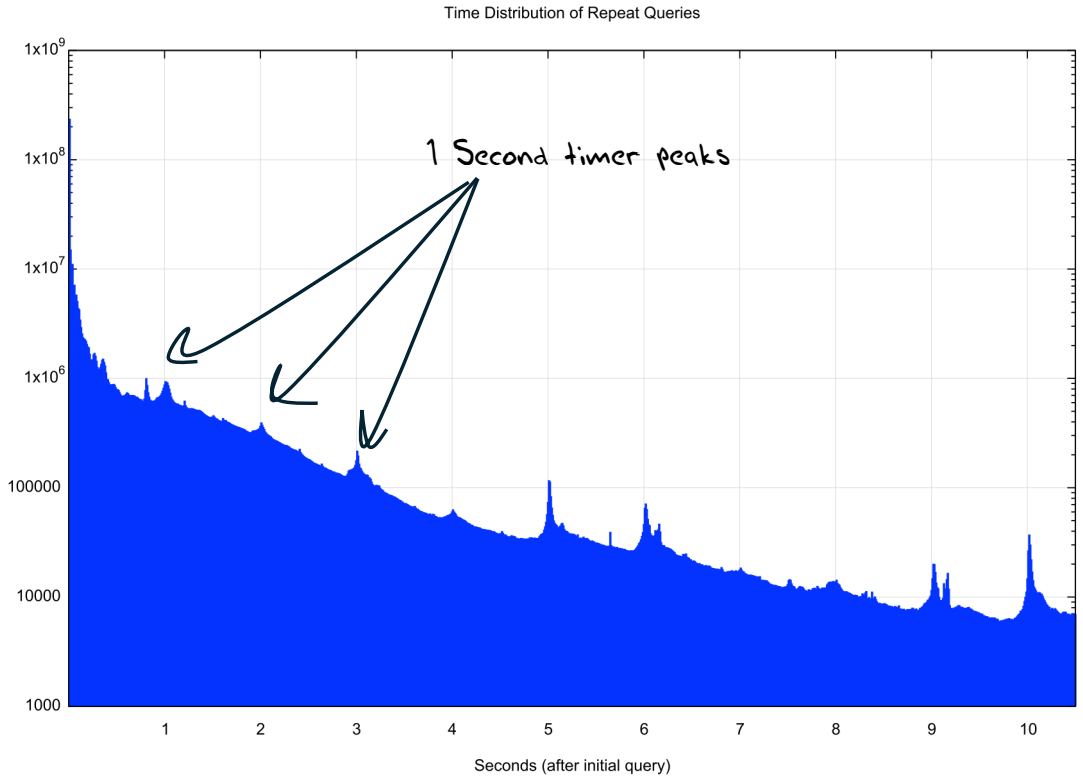




* A “repeated” query has the same Qname and Qtype as a prior query

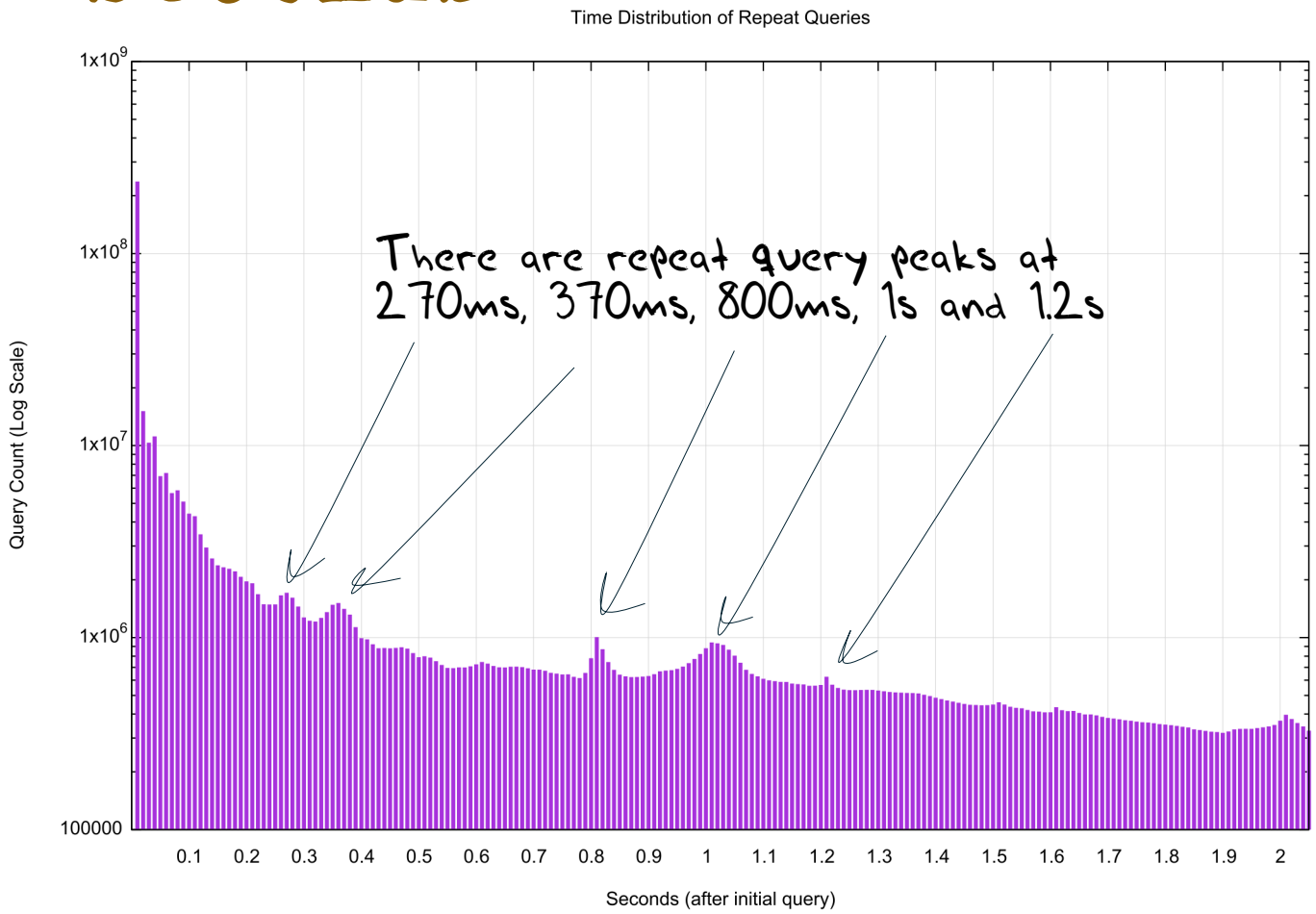
Repeat Query Timing





First 10 seconds

Repeat Query Timing - First 2 Seconds



Repeat Query Timing

- There appear to be two components in the query repeats:
  - **60% of repeated queries are sent within 100ms of the first query**
    - This is a very short time interval, and is generally lower than a default UDP timeout interval
  - The remainder of repeated queries appear to be related to UDP timeout
  - There are observed local peaks at 270ms, 370ms, 800ms, 1s and 1.2s
    - These intervals are more likely to correlate to resolvers' UDP timeouts

60% of repeated queries are sent within 100ms of the first query

- So why is the same query (same query name, same query type) being repeated in this rapid-fire pattern?
  - We looked at individual recursive resolvers in a controlled context and could not reproduce this behaviour from individual recursive resolvers
  - What is so “special” about DNS resolver environments in ISPs?

60% of repeated queries are sent within 100ms of the first query

- So why is the same query (same query name, same query type) being repeated in this rapid-fire pattern?
  - We looked at individual recursive resolvers in a controlled context and could not reproduce this behaviour from individual recursive resolvers
  - What is so “special” about DNS resolver environments in ISPs?
  - **Front End DNS load balancers!**
  - **A possible explanation is anomalous behaviour by DNS front ends that generate spurious rapid-fire DNS query repeats**

It's not just NXDOMAIN responses

- The DNS provides a number of ways for an Authoritative Server to say "No!"
- Possible negative responses from an Authoritative Server:
  - **NXDOMAIN** – *this name does not exist*
  - **NOERROR/NODATA** – *the name exists, but not this Query Type*
  - **SERVFAIL** - *I can't answer you!*
  - **REFUSED** – *I won't answer you!*
  - **Silent Drop** – *Nobody home at this address!*
- How efficient are each of these negative responses in terms of DNS resolver behaviour and query duplication?
- Do we see similar query replication behaviour when the server provides a positive response?

Saying "No!" in the DNS

	Tests	Queries	Ratio	Repeats	Ratio
Control	263,466,799	1,405,946,810	5.34	712,036,505	2.7
NXDOMAIN	115,799,726	512,740,759	4.43	294,857,613	2.55
NODATA	116,158,474	655,772,281	5.65	378,254,000	3.26
SERVFAIL	141,276,753	11,960,521,219	84.66	11,702,543,433	82.83
REFUSED	116,088,436	1,422,926,898	12.26	1,148,833,809	9.9
NO RESP	116,196,452	9,811,833,135	84.44	9,591,903,095	82.55

Control – Always answer!

Repeats is the count of queries with the same Query Name and Query Type

Ratio is the proportion of repeat queries to the number of tests

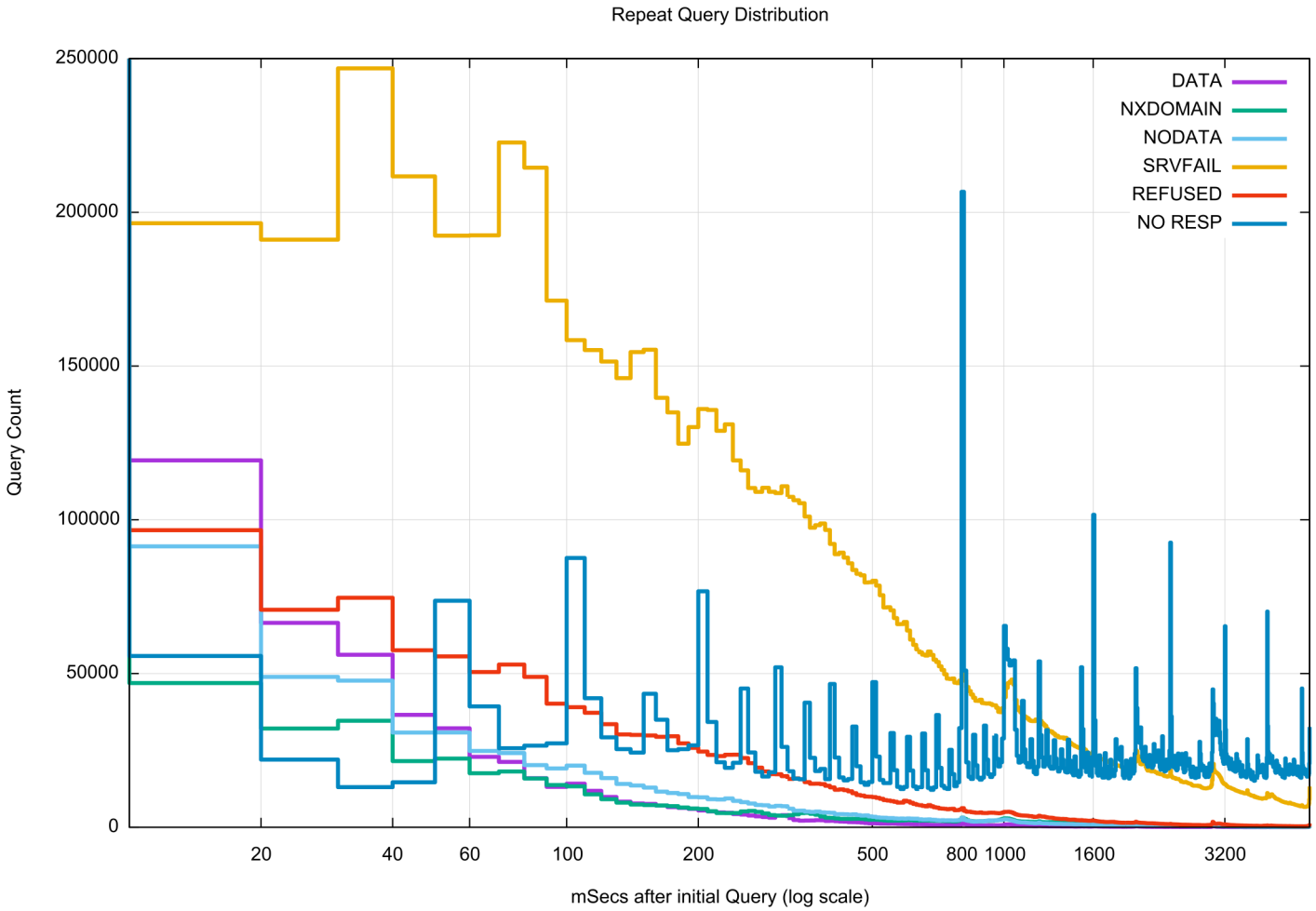
Saying "No!" in the DNS

	Tests	Queries	Ratio	Repeats	Ratio
Control	263,466,799	1,405,946,810	5.34	712,036,505	2.7
NXDOMAIN	115,799,726	512,740,759	4.43	294,857,613	2.55
NODATA	116,158,474	655,772,281	5.65	378,254,000	3.26
SERVFAIL	141,276,753	11,960,521,219	84.66	11,702,543,433	82.83
REFUSED	116,088,436	1,422,926,898	12.26	1,148,833,809	9.9
NO RESP	116,196,452	9,811,833,135	84.44	9,591,903,095	82.55

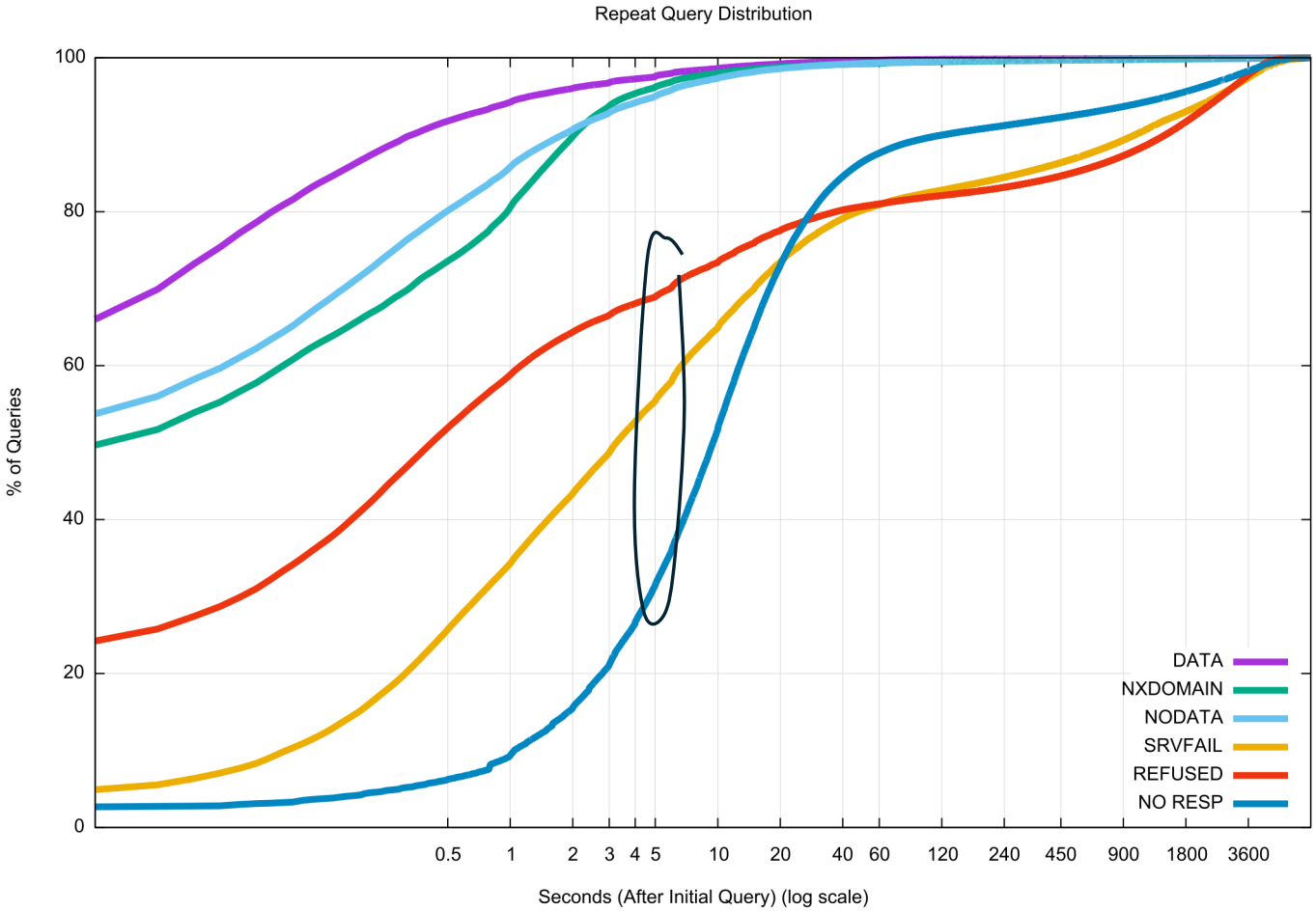


SERVFAIL is as bad as not answering at all!
NXDOMAIN is more efficient than NODATA
REFUSED is less of an invitation to re-query as compared SERVFAIL

Repeat Query Timer Distribution



Query Persistence after 5 seconds



REFUSED, SRVFAIL and NORESPONSE query count is below 70% of total queries for each query type after 5 seconds

If you need to say "No"

- NXDOMAIN is the most effective way of shutting down repeat query behavior
- NOERROR/NODATA is almost as good in terms of query repetition
- REFUSED has a higher query repetition – about 3 – 4 times the repetition rate if NXDOMAIN
- SERVFAIL is about the same as not responding at all! Resolvers are very persistent in re-querying when given a SERVFAIL response

Observations

The DNS query rate per resolved query name is increasing over time. Some potential factors are:

- Changing the Query Type Profile
  - The increasing deployment of IPv6 is echoed in the increasing rate of AAAA queries
  - The adoption of the HTTPS service query is echoed in the rate of HTTPS queries
- Rapid Query Duplication
  - It's unlikely this is due to anomalous behavior of stand-alone recursive resolvers – a more likely cause may be the interaction between load balancing DNS front ends and their resolver farms
- The use of shorter UDP timers
  - Which is intended to improve DNS response times when the original query/response is lost
  - This appears to be coupled with increased use of passing a query to larger pools of recursive resolvers, with shorter timeouts

Just random Musing…

- The DNS appears to operate with an overall query repetition factor of between 3 and 5 when using a UDP transport
- Previous studies have reported that shifting a DNS query load from UDP to TCP drops peak query performance in the server to some 30% of achievable UDP levels
- If the repetition factor when using TCP was eliminated * then this leads to the thought that if the DNS used TCP rather UDP, then would the overall system performance be about the same?

* A hypothetical TCP-only DNS platform (and TCP-only clients) is challenging to test at scale within in today's operational Internet!

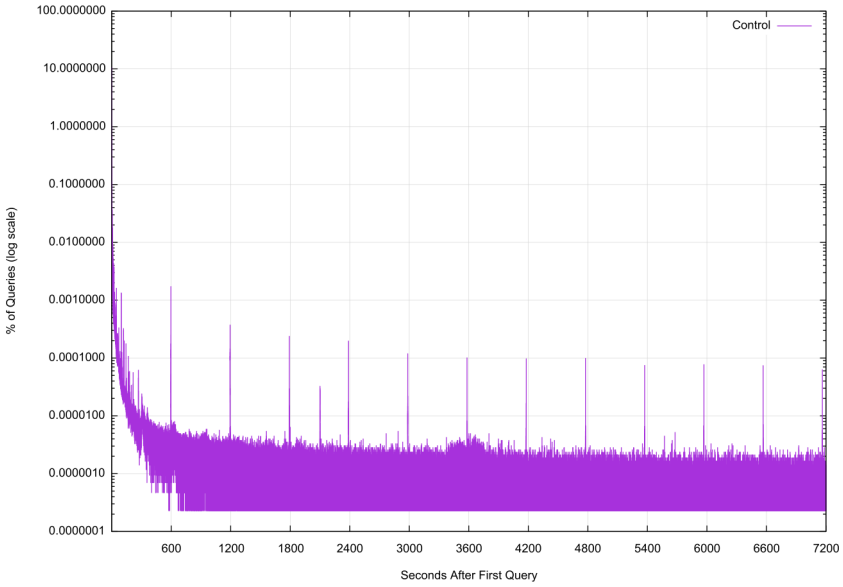
Thanks!

Questions?

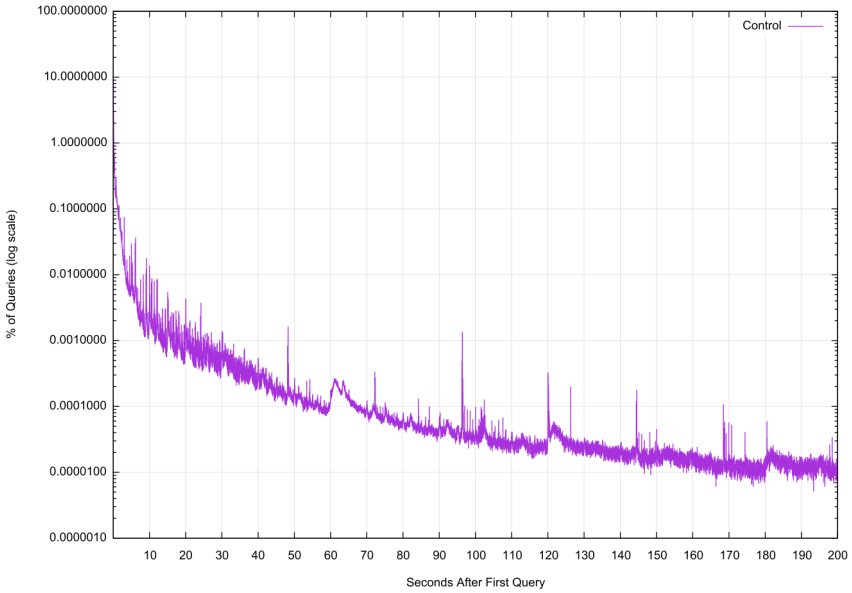
Additional Material

Duplicate Profiles

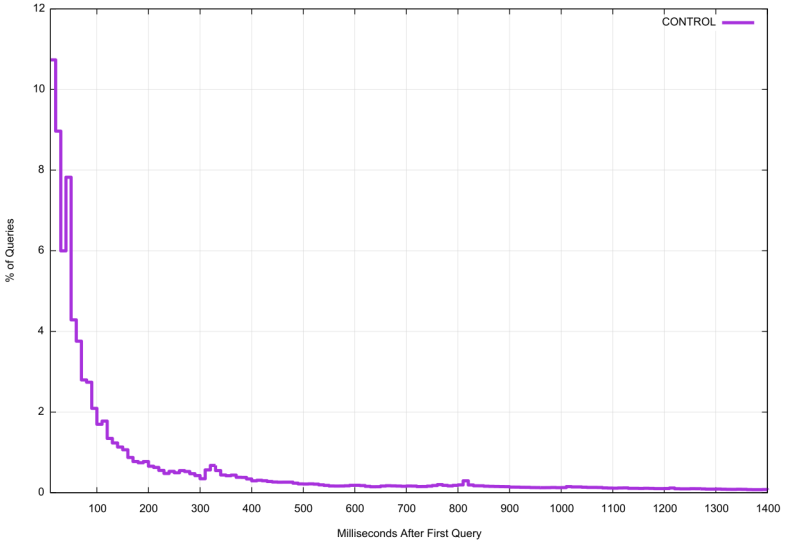
CONTROL



Dups - 2 hours



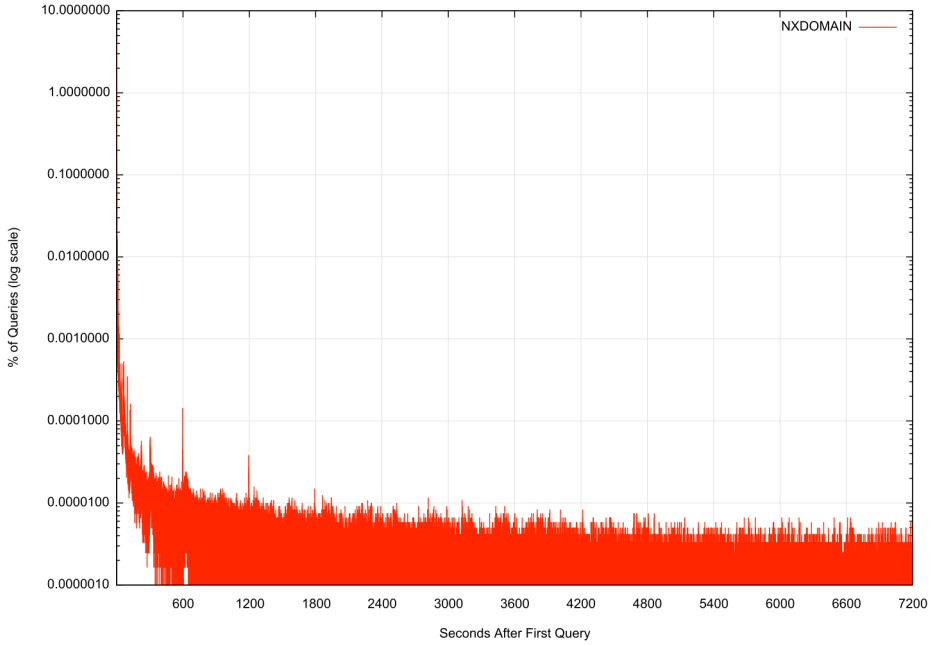
Dups - 200 secs



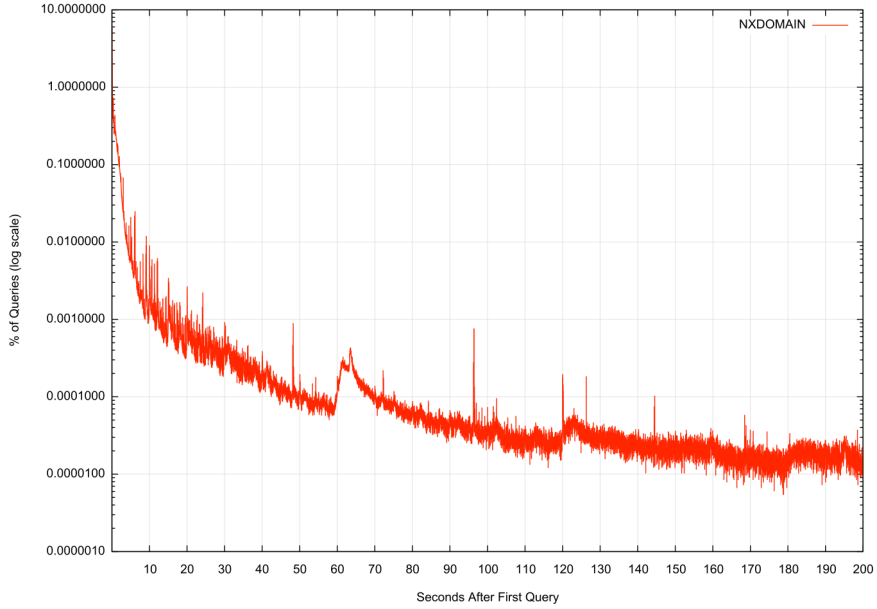
Dups - 1.4 secs

Duplicate Profiles

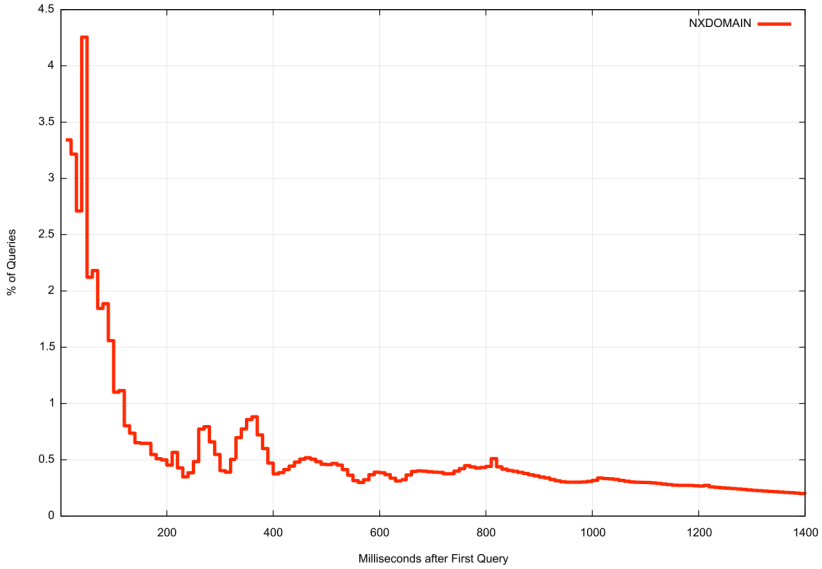
NXDOMAIN



Dups - 2 hours



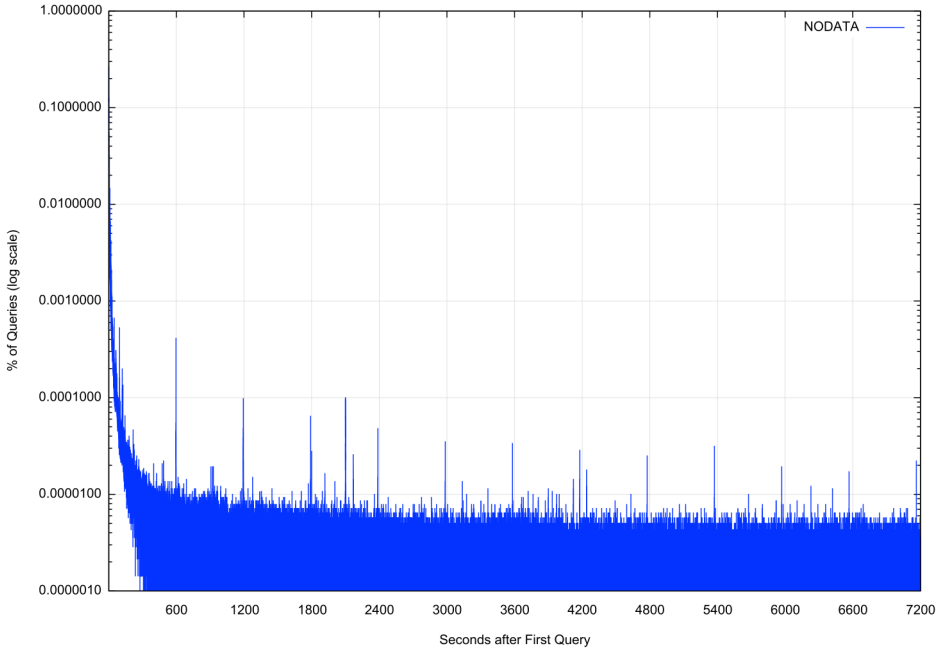
Dups - 200 secs



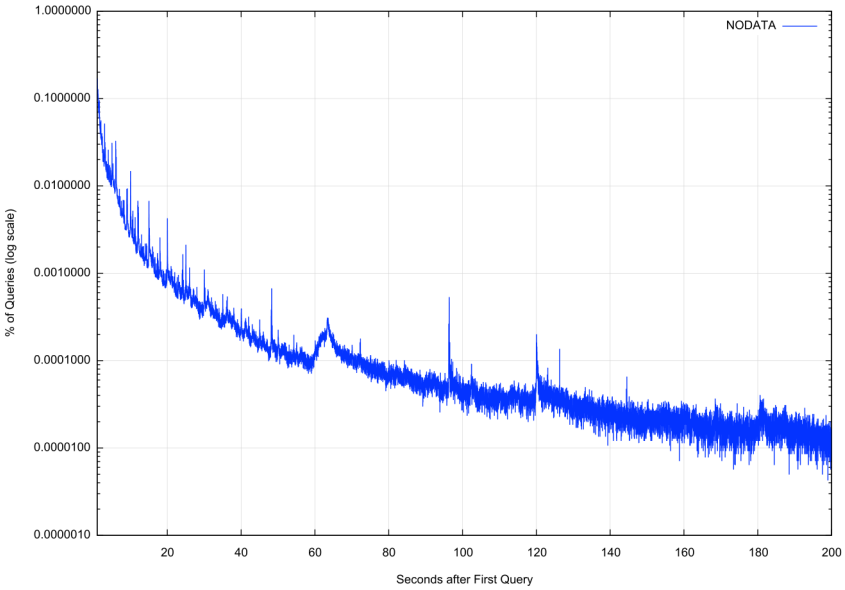
Dups - 1.4 secs

Duplicate Profiles

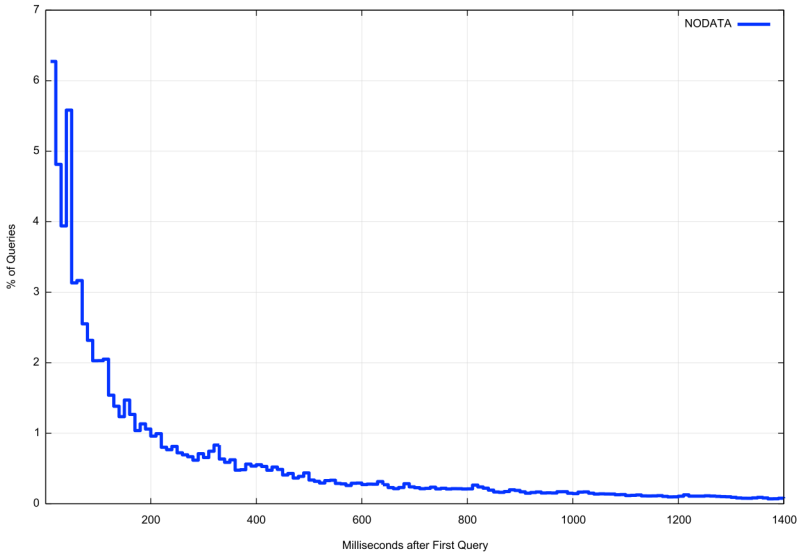
NOERROR/NODATA



Dups - 2 hours



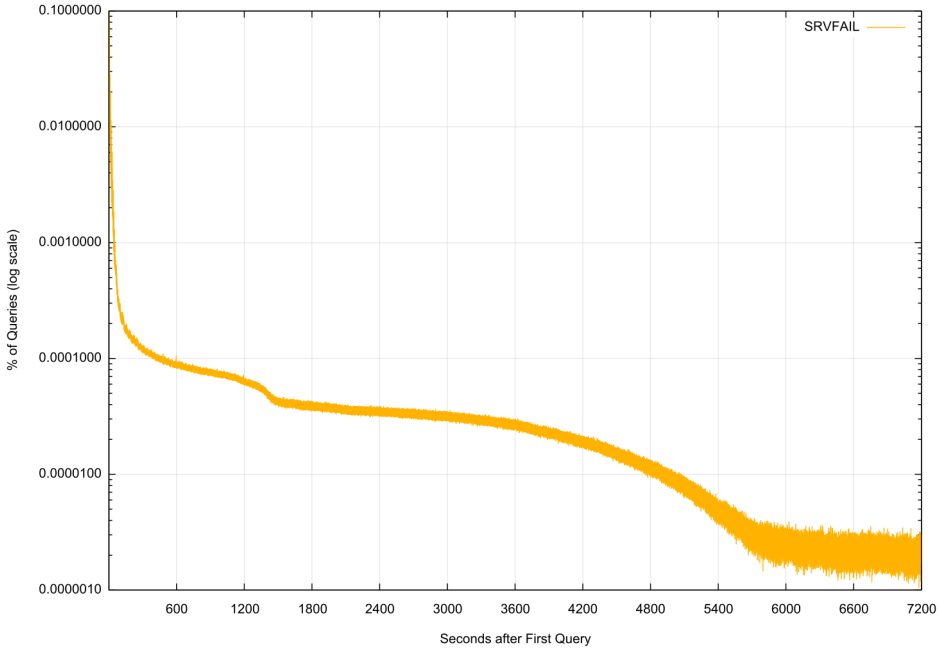
Dups - 200 secs



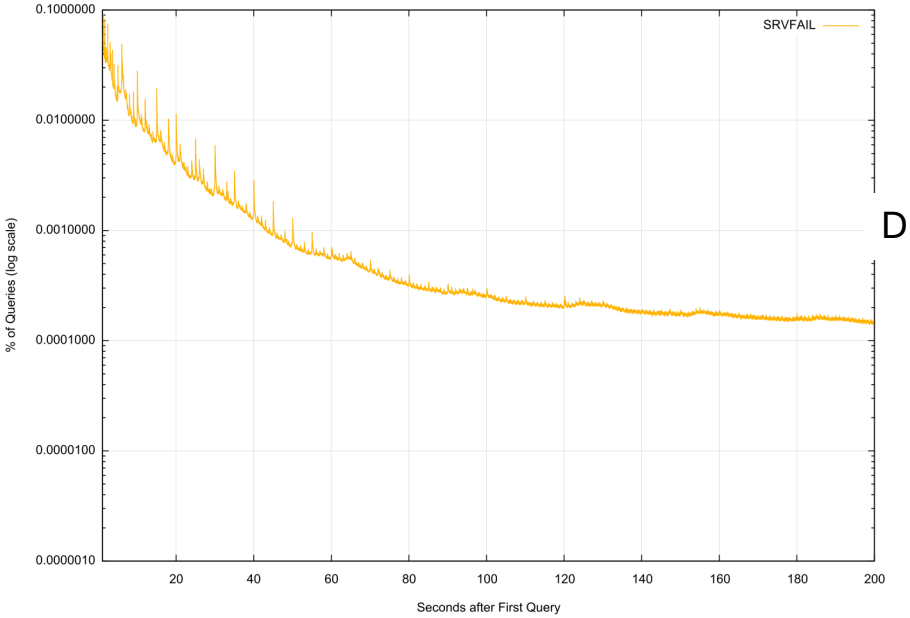
Dups - 1.4 secs

Duplicate Profiles

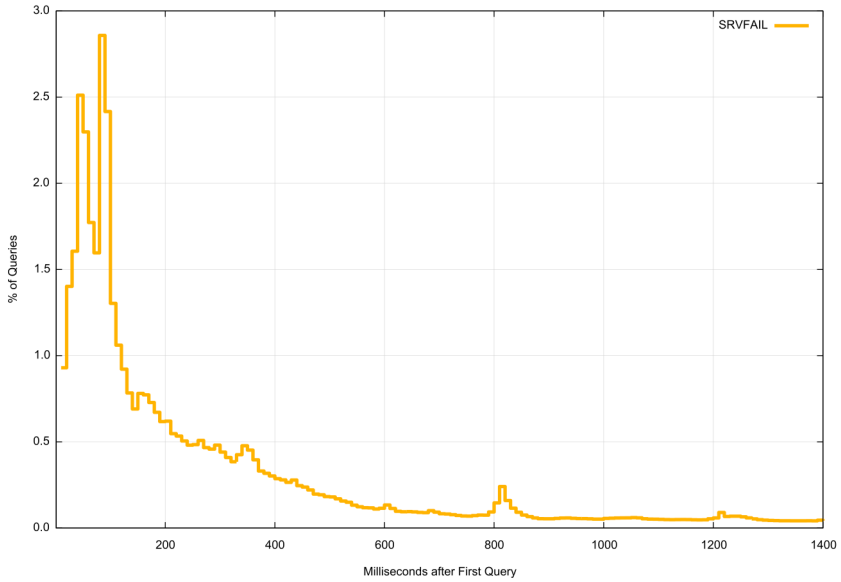
SERVFAIL



Dups - 2 hours



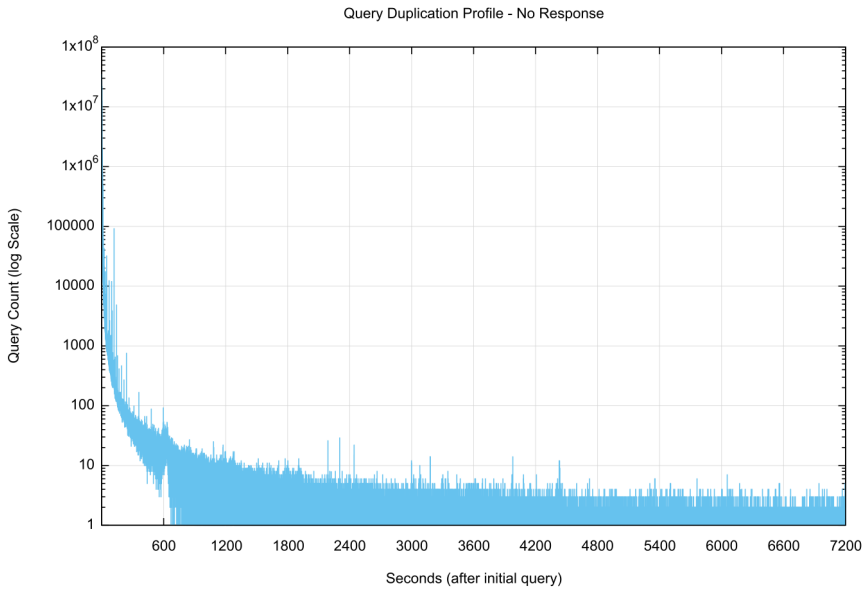
Dups - 200 secs



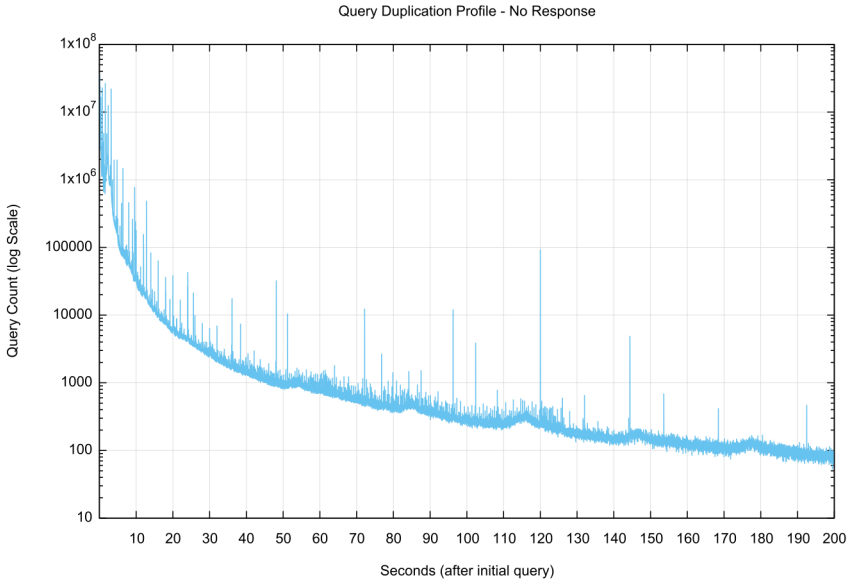
Dups - 1.4 secs

Duplicate Profiles

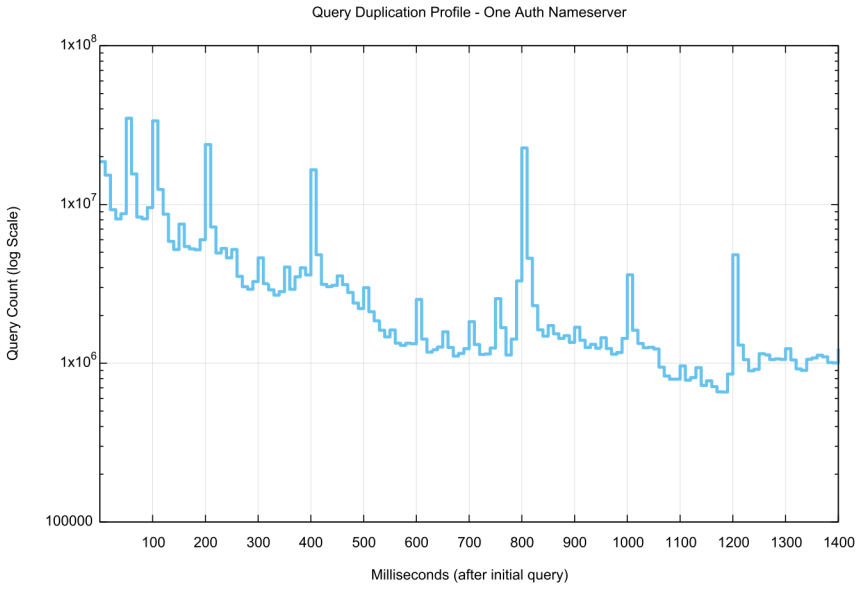
NO RESPONSE



Dups - 2 hours



Dups - 200 secs



Dups - 1.4 secs

Control: Saying "Yes"

- To provide a benchmark for DNS resolution behaviour, let's look at the queries seen by an authoritative nameserver when it positively answers A and AAAA queries

  (The name is provisioned with A and AAAA records, but no HTTPS record)

Tests	267,461,399
Total Queries	1,390,651,435
A	787,422,754
AAAA	350,104,101
HTTPS	253,124,580
Resolvers	766,849,965

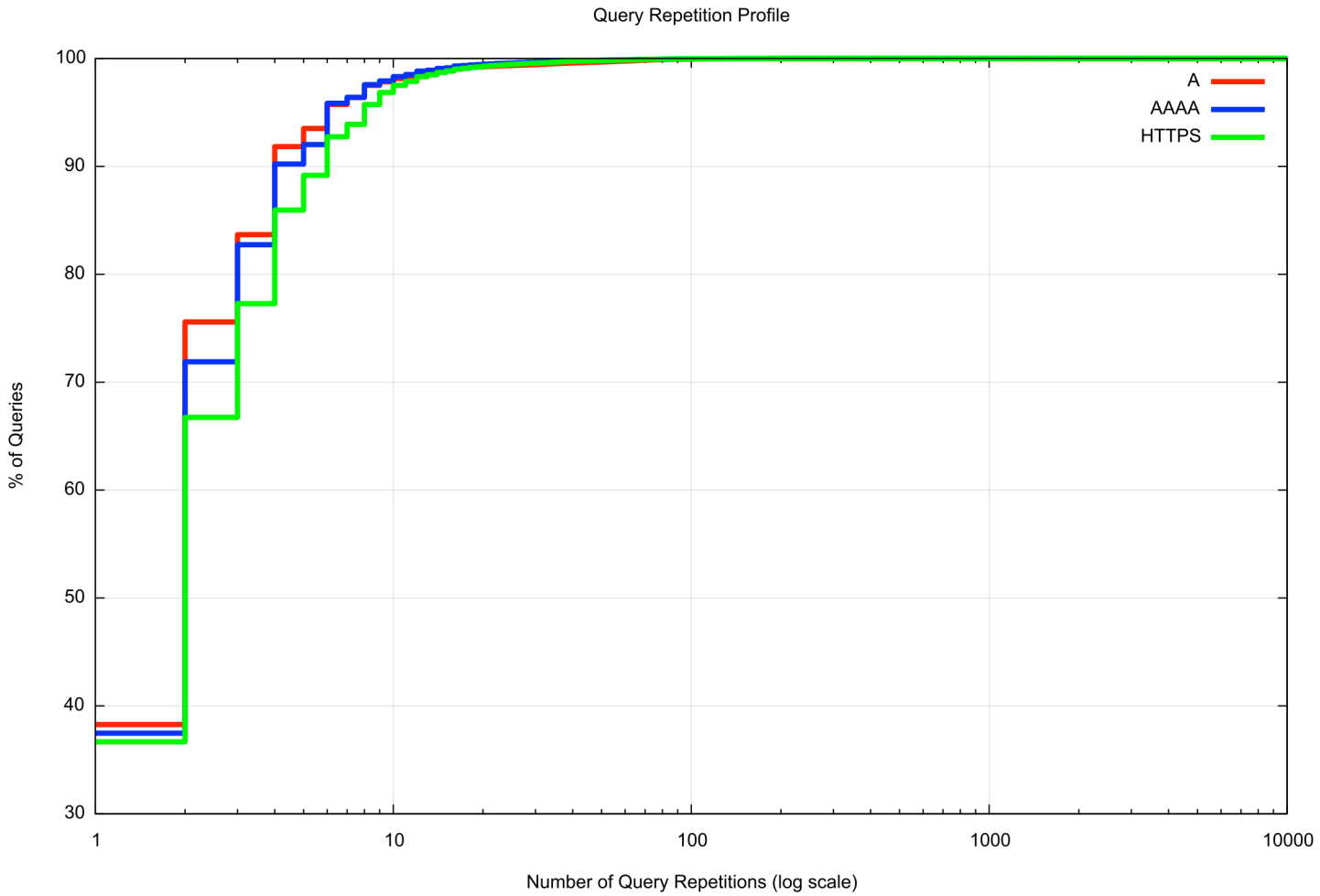
Data from a 1 week run across 5/8 through 11/8 2026

Saying "Yes!"

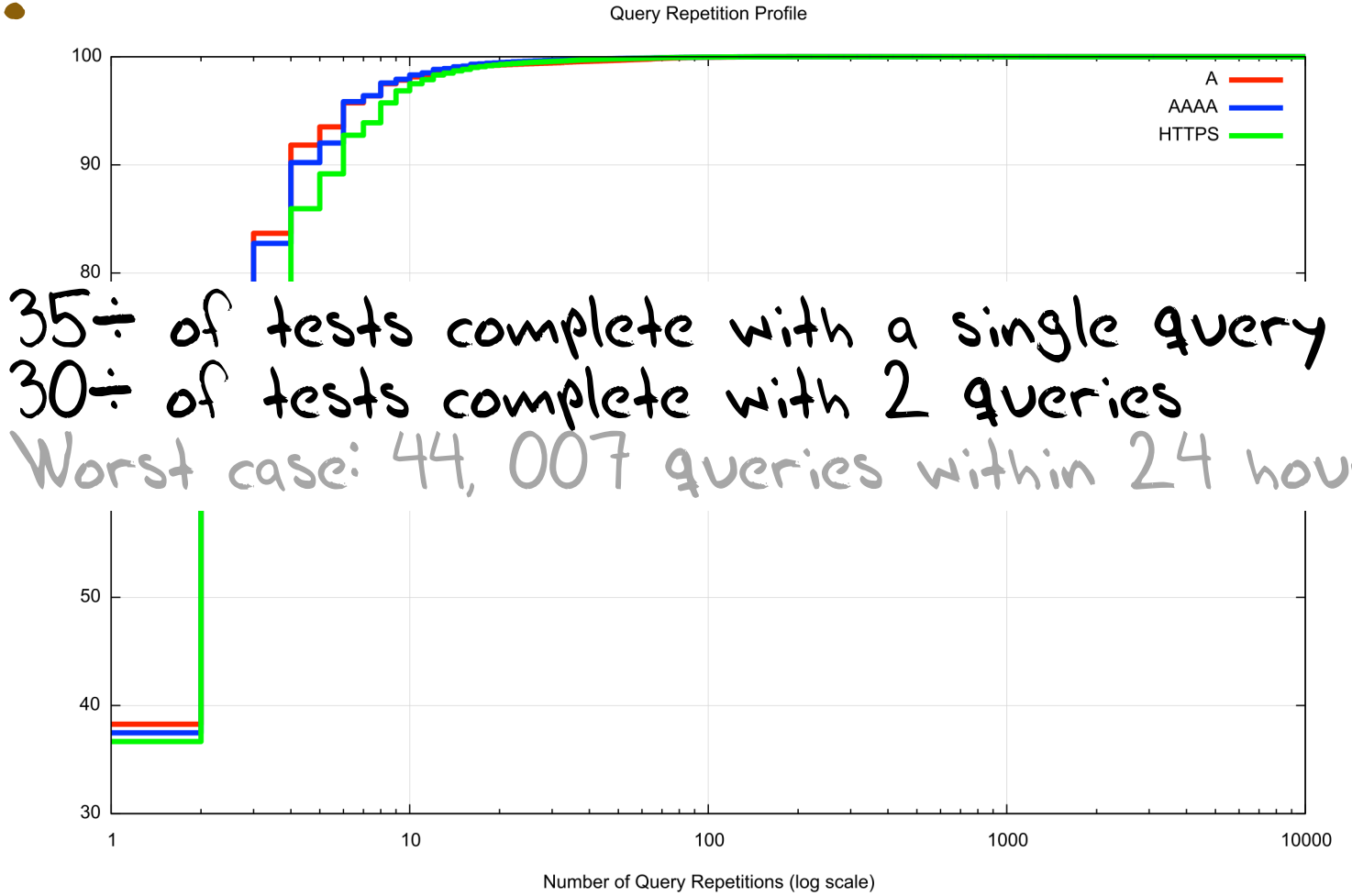
Repeat Queries by Query Type

Q Type	Tests	Queries	Repeats
A	266,100,136	787,422,754	2.96
AAAA	133,656,855	350,104,101	2.62
HTTPS	95,097,186	253,124,580	2.66

Query Repetition Profile - Yes!



Query Repetition Profile - Yes!



35÷ of tests complete with a single query

30÷ of tests complete with 2 queries

Worst case: 44,007 queries within 24 hours!