The Different Ways of Minimizing ANY

Observations on *Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY* (a.k.a. RFC 8482)

Edward Lewis

DNS-OARC Workshop 32
8 February 2020

Starting Points

- Why "*Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY*"?
- A Measurement/Observation on its Impact
- Surveying Implementations
- Underlying Principles
- The Need for Increasing Simplicity
- Relationship of Protocol Development, Code Development and Operations

Providing Minimal-Sized Responses to ... QTYPE=ANY (The set up)

- QTYPE=ANY
  - Benign: snooping on a domain name at an authoritative server
  - Problematic: expecting multiple sets (A and AAAA) in one lookup
  - Malicious: a message-size amplifier from a well-provisioned source

- Stop the bad use while softening the blow for the good use
  - Hard fails (RCODE="bad") drive traffic up or waste a round trip
  - DNS has no clear, polite response for "no!", especially "not anymore!"
  - Protocol developers tried to appease everyone

How does the document specify "saying no"

- (4.1) Answer with a Subset of Available RRsets
  - ...MAY consist of a single RRset owned by the name specified in the QNAME
- (4.2) Answer with a Synthesized HINFO RRset
  - If there is no CNAME present at the owner name matching the QNAME
- (4.3) Answer with Best Guess as to Intention

Providing Minimal-Sized Responses to ... QTYPE=ANY (results)

- Result in RFC 8482 (incomplete, out of context quotes):
  - (4.1)*...This mechanism does not signal ... that an incomplete subset ... has been returned.*
  - (4.2) *A system that receives an HINFO response SHOULD NOT infer ..., it is not possible to tell with certainty whether the HINFO RRset received was synthesized.*
  - (4.3) *In some cases, it is possible to guess what the initiator wants in the answer (but not always).*

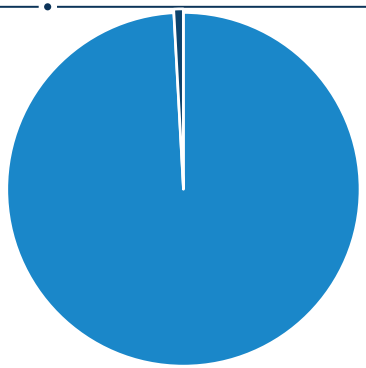
Clarifying my "Complaint"

- It's good to limit or eliminate QTYPE=ANY and good to maintain backwards compatibility
- My concern is that the document does so by increasing non-determinism in the protocol
  - Increasing complexity?
- Larger cloud overhead:
  - This isn't the only time this has happened
    - Overloading the meaning of RCODE=SERVFAIL (for DNSSEC)
    - Overloading the TXT record (SPF or TXT for mail)

How Has Minimizing ANY Played Out?

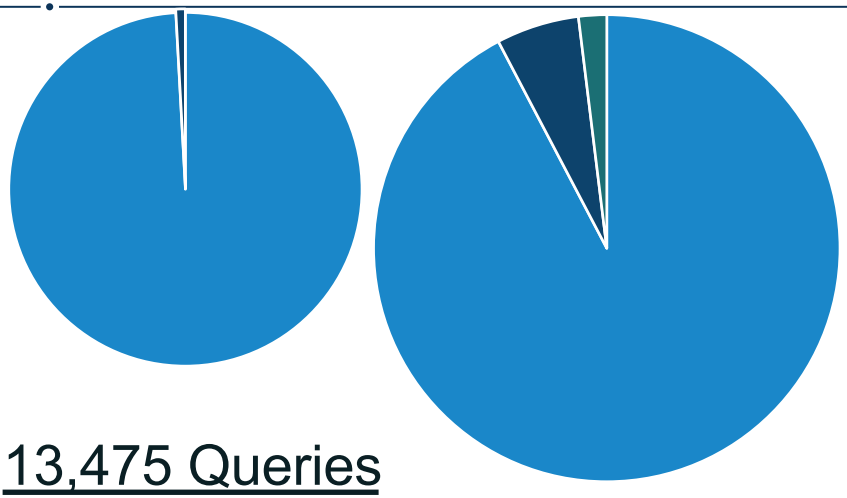
- A small experiment (17 Jan 2020) covering nameservers for the Top-Level Domain registries
  - For convenience, figuring TLD servers are well-managed resources
- 13,475 queries over UDP and 13,475 more over TCP
  - For UDP: 260 contained a "minimized ANY answer" 10 Different Ways!
  - For TCP: 251 contained a "minimized ANY answer" 9 Different Ways!
- Notes on these numbers: there is some double counting of "decisions" as some IP addresses behave the same way for multiple zones
  - IP addresses behaved differently depending on the zone.

UDP Responses (1)



13,475 Queries
13,358 Responses
117 No Response

UDP Responses (2)



13,475 Queries

13,358 Responses

117 No Response

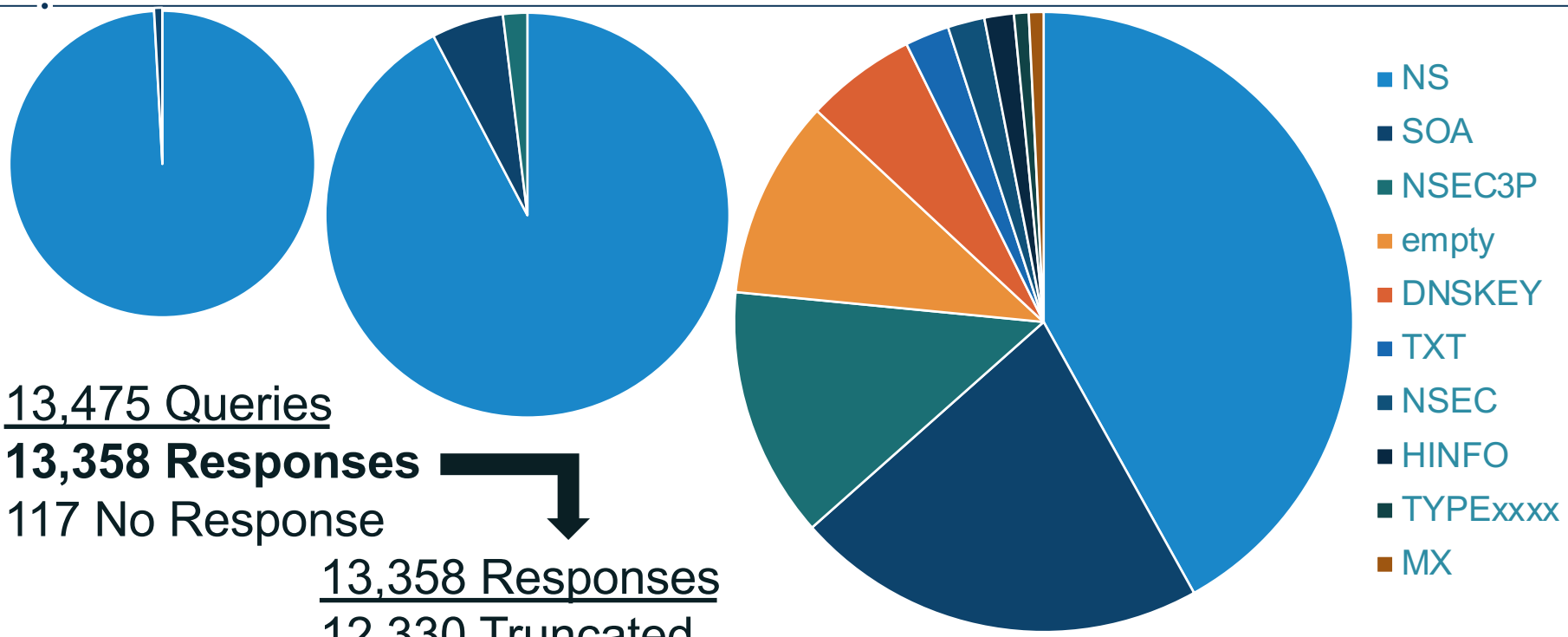
13,358 Responses

12,330 Truncated

768 "Valid" responses

260 Other Responses

UDP Responses (3)



13,475 Queries

13,358 Responses

117 No Response

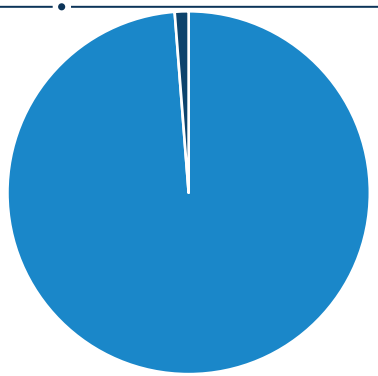
13,358 Responses

12,330 Truncated

768 "Valid" responses

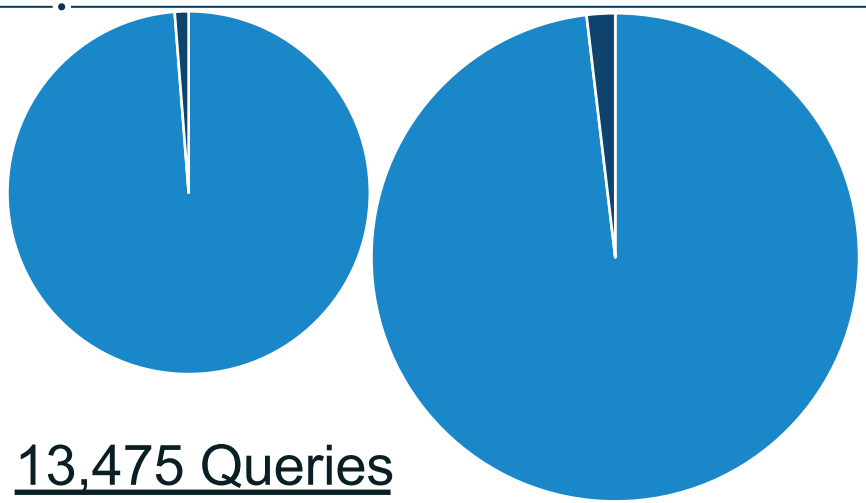
260 Other Responses → **10 different ways**

TCP Responses (1)



13,475 Queries
13,309 Responses
166 No Response

TCP Responses (2)



13,475 Queries

13,309 Responses

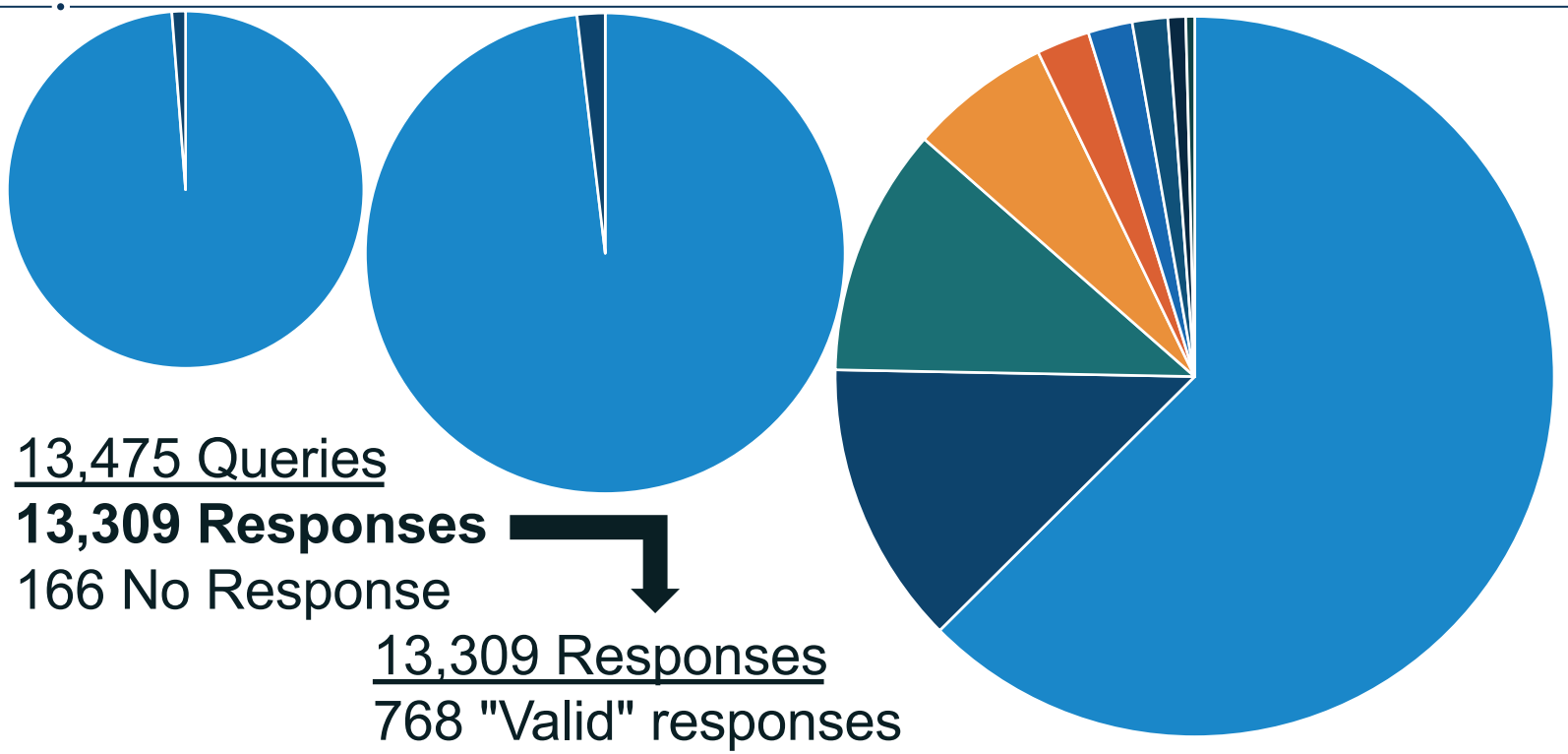
166 No Response

13,309 Responses

768 "Valid" responses

260 Other Responses

TCP Responses (3)



13,475 Queries

13,309 Responses

166 No Response

13,309 Responses

768 "Valid" responses

260 Other Responses → **9 different ways**

Drilling into the Numbers

- Value magnitudes are not terribly meaningful
  - Servers may share IP addresses and serve multiple TLDs
    - Hence double counting of what "large DNS operators" do
  - Looking at "denials" per IP address shows that some IP addresses alter their "way" of saying no
    - It is operator choice or implementation dependent?
    - Is it a per zone option or server option?
- For at least one IP address
  - The way "no" is said differs from zone to zone

The "Big Four" Open Source Implementations

Implementation - UDP	Empty Answer	NS-only	Other only Types
Implementation 1	Yes (10)	Yes (5)	SOA(1)
Implementation 2	No	Yes (6)	
Implementation 3	Yes (4)	Yes (84)	DNSKEY(7), NSEC3PARAM (7), TXT (6), NSEC(5), MX(2), SOA(1)

Implementation - TCP	Empty Answer	NS-only	Other only Types
Implementation 1	Yes (10)	Yes (8)	
Implementation 2	No	Yes (3)	
Implementation 3	Yes (4)	Yes (87)	DNSKEY(8), NSEC3PARAM (7), TXT (6), NSEC(5), MX(2)

- Magnitudes are not terribly meaningful

How are NSD and BIND Configured?

- NSD 4.2.3:
  - *refuse-any: <yes or no>*
    - *...sends truncation in response to UDP type ANY queries, and it allows TCP type ANY queries like normal... The default is no.*
- BIND 9.14.6:
  - *minimal-any*
    - *over UDP, the server will reply with only one of the RRsets (first one found ... not necessarily the smallest...). The default is no.*

How are Other Implementations Configured?

- Knot and PowerDNS:
  - Couldn't find documentation showing how to configure it
  - Did find some email denying it is implemented

- Observed Behaviors
  - Don't seem to agree with the configuration documentation
  - or maybe the strings in "version.bind" aren't accurate

What Does This Mean?

- I'm a bit baffled by this
- Can't see evidence that operators are making, or could make, the choices specified in the protocol modification document (RFC)
  - There is evidence of the synthesized HINFO option (but not from a server identifying its code base)
  - Don't see how servers respond differently based on QNAME (but they do)

Two Reasons Why This Bothers Me So

- A principle of protocol design
- Observation about levels of staff expertise

A Protocol Ought to be Described by a State Machine

- States of communication ought to be well known, understood, and secure on both sides of a channel
  - Definitive transitions between states based on transmissions and timeouts
  - Each side expects specific reaction(s) to its transmission

- The DNS is already a poor model of this
  - In my younger days I tried to build a state machine and failed

How does this apply to my observations

- I'm asking for QTYPE=ANY at TLD Apex names served by authoritative servers
  - I have an expectation of what will be there
    - SOA, NS
    - Maybe a set of DNSSEC record sets for NSEC3
    - Or maybe a set of DNSSEC record sets for NSEC
    - Maybe others
  - So far, I am able to detect when a server is minimizing ANY via other means
- But in the general case (non apex), I can't tell clearly

Should I be able to detect a minimized ANY response?

- This is a fair question
  - If I know what I want, it would be better to ask for it (in parallel)
  - The approach we have is pragmatic on many levels
- But my concern is about the protocol design process
  - For the sake of a state machine model, determinism is desired
- In the long run, pragmatic short cuts lead to technical debt
  - Perhaps we've lost the battle already

Another concern: Staff Expertise

- Based on an experience
  - Network Operations Center staff mean skill level is trending down
    - Once had a staff member tell me "I don't know how to read a traceroute"
  - We promote people, we expand staff, we grow coverage
  - It's inevitable
- What should we do?
  - Make the protocol simpler, not more complex
  - This enables better tooling, automation, etc.

Gaps

- Protocol Engineers describe ways software can be written, with an expectation that operators will be able to cope with that
  - More general solutions, built around assumptions of operations
- Operators have a myriad of issues to juggle, with avoiding "tickets" of utmost importance
  - A need to lean on pre-packaged software to perform duties
- Software Developers are in the middle of this
  - https://ietf.org/blog/herding-dns-camel/

Classic DevOps

- Protocol Engineers
  - Maximize Functionality
- Operators
  - Minimize Downtime

What Do We Do About This?

- As software developers
  - Do what can be done to improve what comes out of the IETF
  - Comment on documents describing protocol enhancements
- As operators
  - At what cost "backwards compatibility?"
  - Learn to deal with changed behaviors/changed defaults
- As researchers
  - Quantify impact of protocol modifications
  - Find the good so we can do more of that

Engage with ICANN

Thank You and Questions

Visit us at **icann.org**
Email: edward.lewis@icann.org

@icann

linkedin/company/icann

facebook.com/icannorg

slideshare/icannpresentations

youtube.com/icannnews

soundcloud/icann

flickr.com/icann

instagram.com/icannorg