# Measuring ATR

Joao Damas
Geoff Huston
@apnic.net
March 2018

# September 2017:

## V6, the DNS and Fragmented UDP Responses

We used the Ad platform to enroll endpoints to attempt to resolve a DNS name that included a IPv6 fragmented UDP response when attempting to resolve the name server's name

Total number of tests: 10,851,323

Failure Rate in receiving a large response: 4,064,356

IPv6 Fragmentation Failure Rate: **38%**

# The Internet has a problem …

- Instead of evolving to be more flexible and more capable, it appears that the Internet's transport is becoming more ossified and more inflexible in certain aspects
- One of the major issues here is the handling of large IP packets and IP level packet fragmentation
- We are seeing a number of end-to-end paths on the network that no longer support the carriage of fragmented IP datagrams
- We are concerned that this number might be getting larger, not smaller

# The Internet has a problem …

- What about the DNS?
  - One application that is making increasing use of large UDP packets is the DNS
  - This is generally associated with DNSSEC-signed responses (such as “dig +dnssec DNSKEY org”) but large DNS responses can be generated in other ways as well (such as “dig . ANY”)
  - In the DNS we appear to be relying on the successful transmission of fragmented UDP packets, but at the same time we see an increasing problem with the coherence in network and host handling of fragmented IP packets, particularly in IPv6

# Changing the DNS?

- But don’t large DNS transactions use TCP anyway?
  - In the original DNS specification only small (smaller than 512 octets) responses are passed across UDP.
  - Larger DNS responses are truncated and the truncation is intended to trigger the client to re-query using TCP
  - EDNS(0) allowed a client to signal a larger truncation size threshold, and assumes that fragmented DNS is mostly reliable
  - But what if it’s not that reliable?

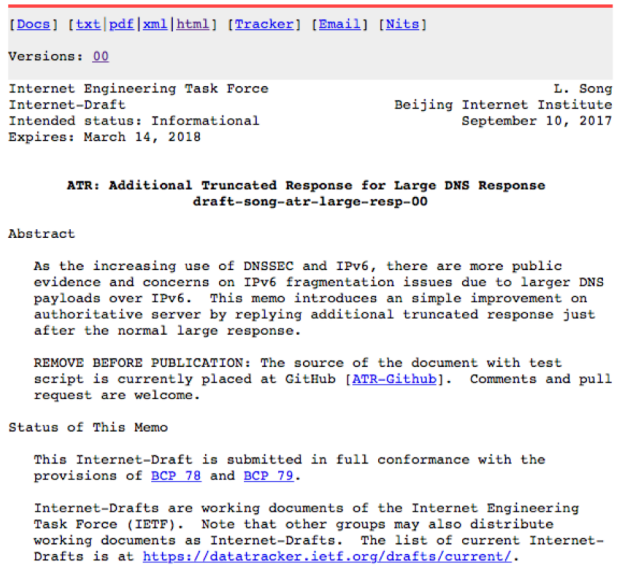
# What is “ATR”?

- It stands for “Additional Truncated Response”

  Internet draft: draft-song-atr-large-resp-00

  September 2017

  Linjian (Davey) Song, Beijing Internet Institute



[Docs] [txt|pdf|xml|html] [Tracker] [Email] [Nits]

Versions: 00

Internet Engineering Task Force — L. Song
Internet-Draft — Beijing Internet Institute
Intended status: Informational — September 10, 2017
Expires: March 14, 2018

**ATR: Additional Truncated Response for Large DNS Response**
**draft-song-atr-large-resp-00**

Abstract

As the increasing use of DNSSEC and IPv6, there are more public evidence and concerns on IPv6 fragmentation issues due to larger DNS payloads over IPv6. This memo introduces an simple improvement on authoritative server by replying additional truncated response just after the normal large response.

REMOVE BEFORE PUBLICATION: The source of the document with test script is currently placed at GitHub [ATR-Github]. Comments and pull request are welcome.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

- It’s a hybrid response to noted problems in IPv4 and IPv6 over handling of large UDP packets and IP fragmentation
- ATR adds an additional response packet to ‘trail’ a fragmented UDP response
- The additional response is just the original query with the Truncated bit set, and the sender delays this additional response packet by 10ms

# The Intention of ATR

Today:

- If the client cannot receive large truncated responses then it will need to timeout from the original query,
- Then re-query using more resolvers,
- Timeout on these queries
- Then re-query using a 512 octet EDNS(0) UDP buffersize
- Then get a truncated response
- Then re-query using TCP

# The Intention of ATR

ATR

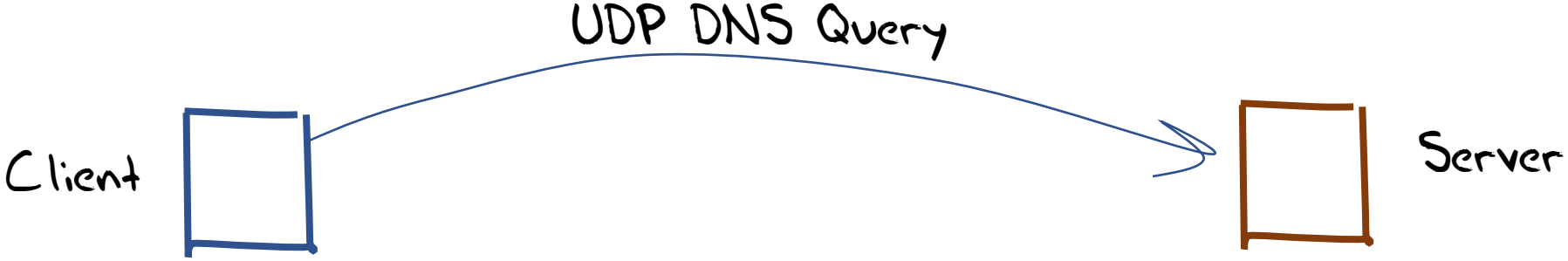
Today:

- If the client cannot receive large truncated responses then it will need to timeout from the original query,
- Then re-query using more resolvers,
- Timeout on these queries
- Then requery using a 512 octet EDNS(0) UDP buffersize
- Then get a truncated response within a few ms
- Then requery using TCP

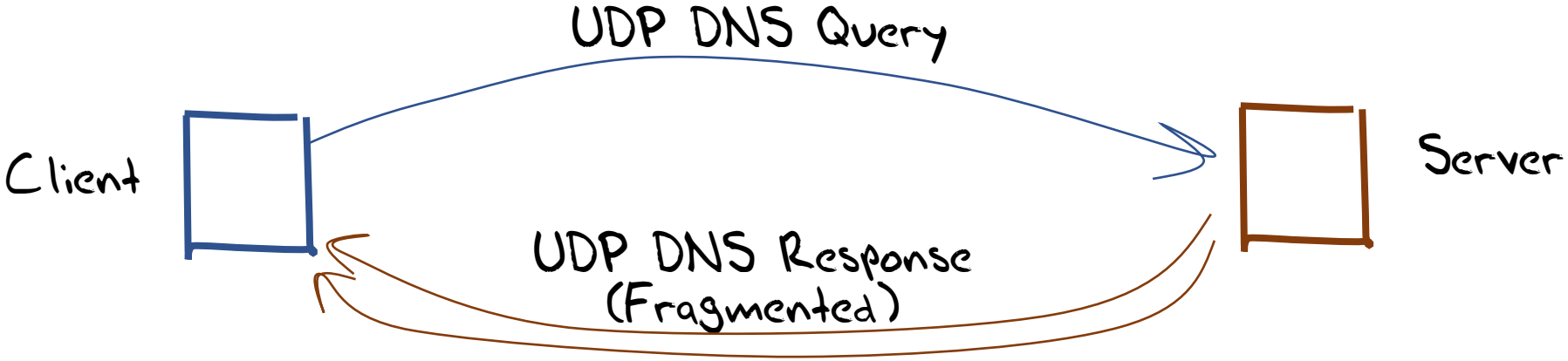
# The Intention of ATR

- When a UDP DNS response is fragmented by the server, then the server will also send a delayed truncated UDP DNS response
  - The delay is proposed to be 10ms
- If the DNS client receives and reassembles the fragmented UDP response the ensuing truncated response will be ignored
- If the fragmented response is lost due to fragmentation loss, then the client will receive the short truncated response
- The truncation setting is intended to trigger the client to re-query using TCP

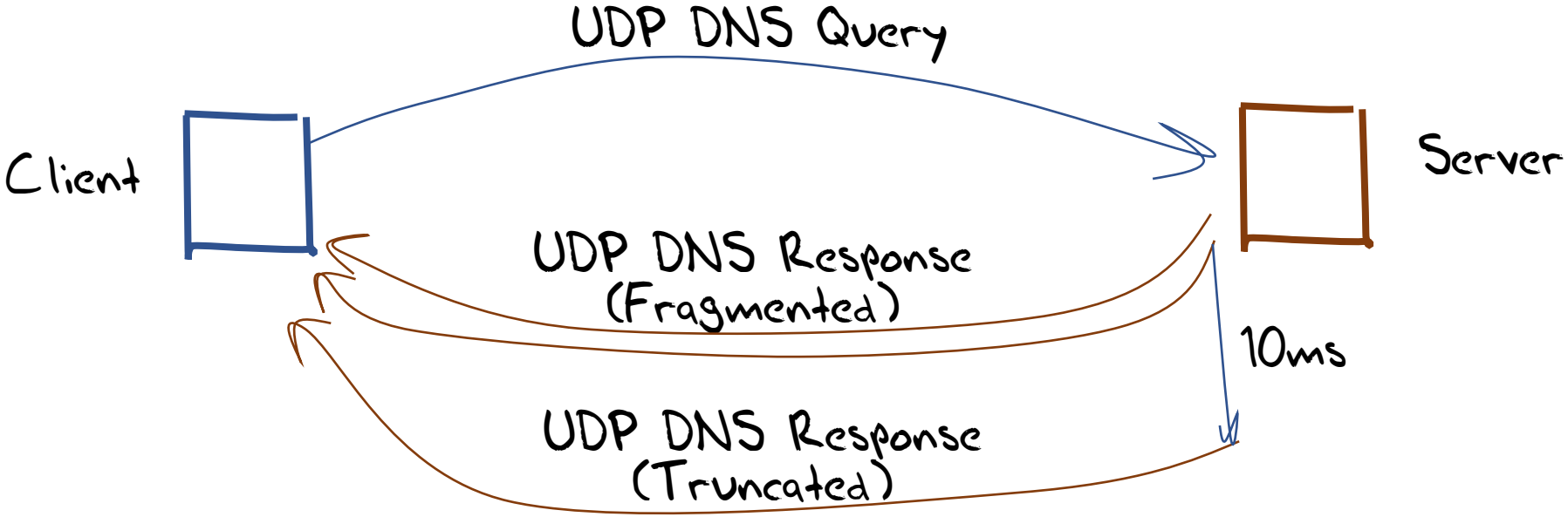
# ATR Operation



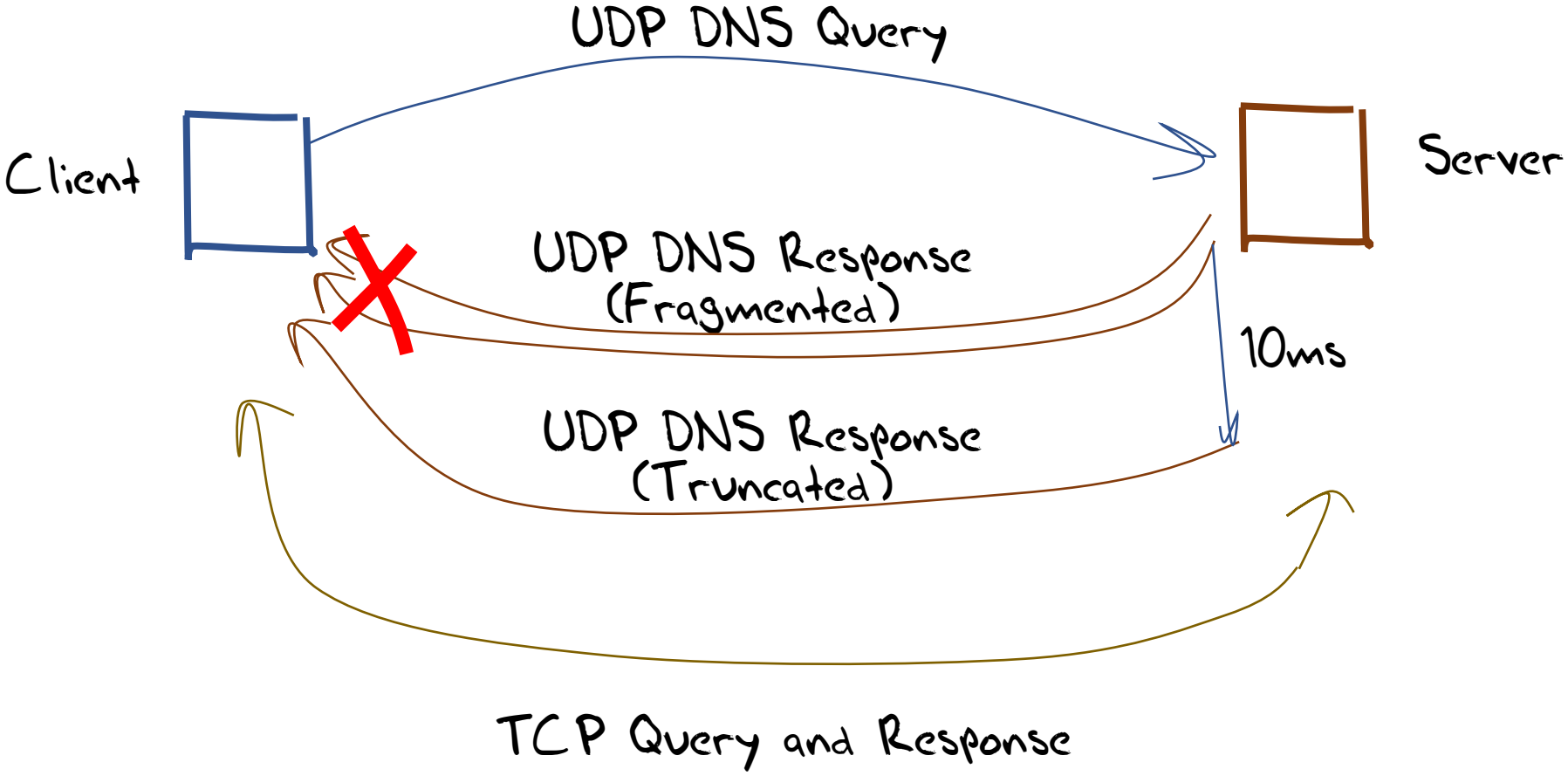
# ATR Operation



# ATR Operation



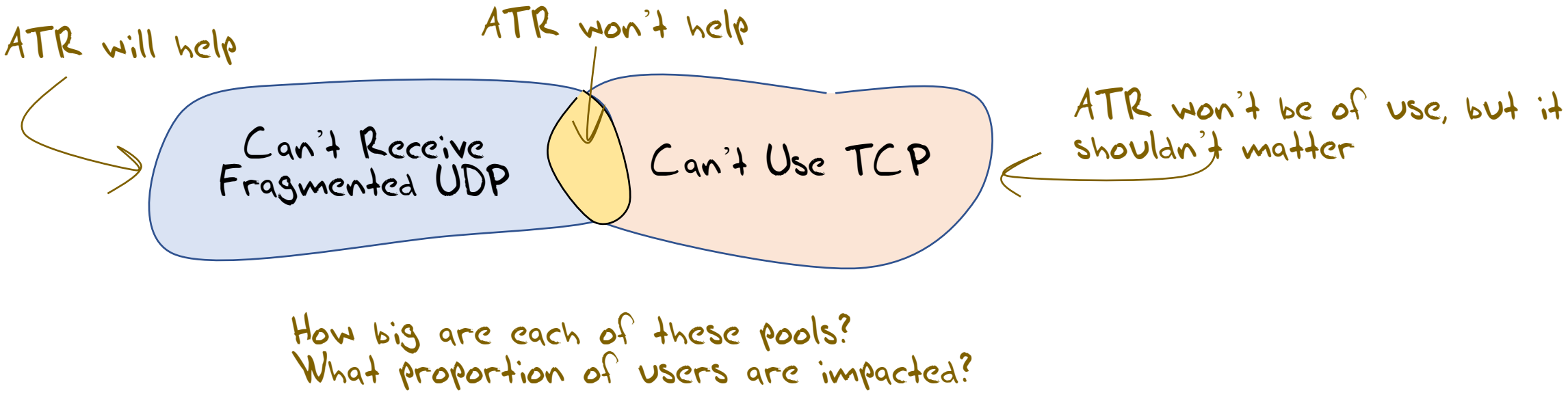
# ATR Operation



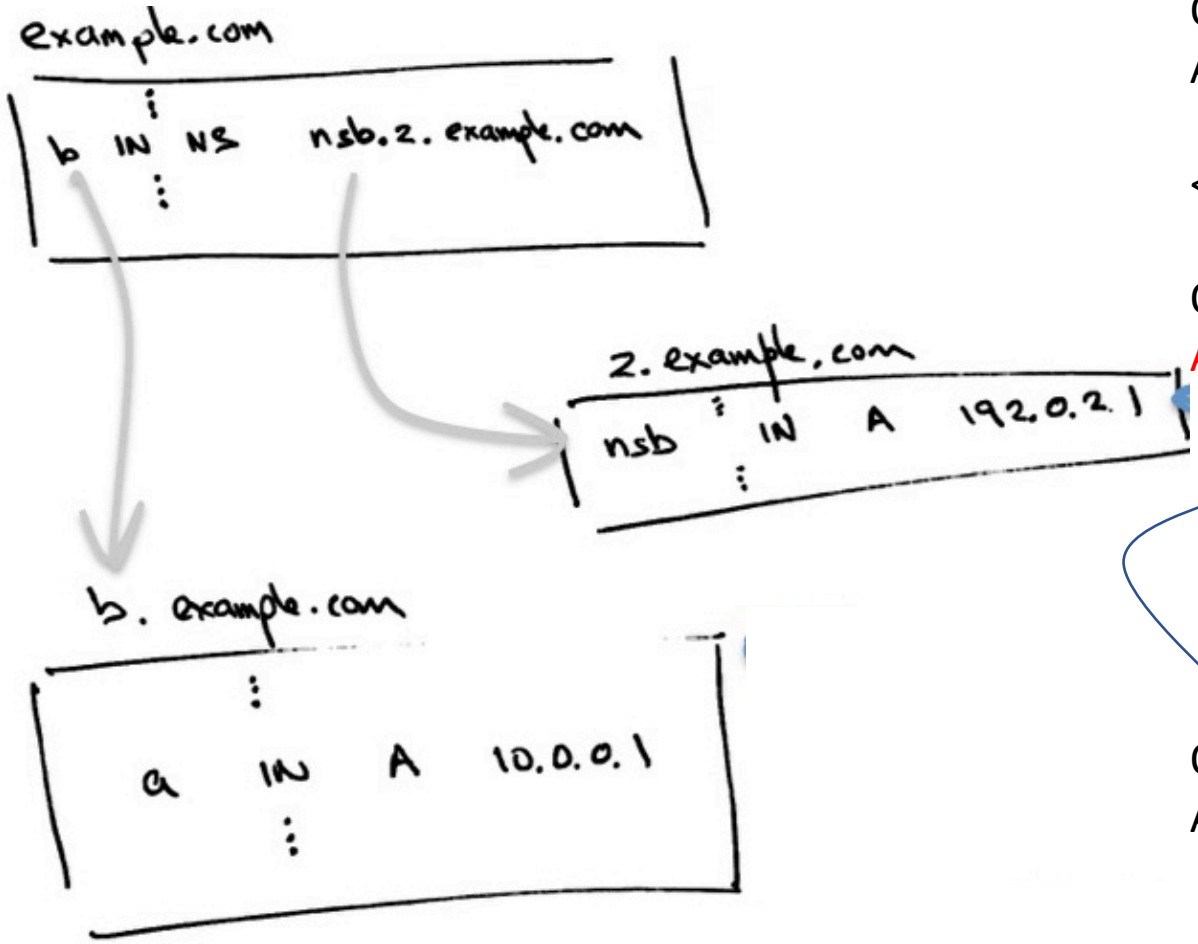
# What could possibly go wrong?

- Network level packet re-ordering may cause the shorter truncated response packet to overtake the fragmented response, causing an inflated TCP load, and the potential for TCP loss to be triggered
- Not every client DNS system supports using TCP to emit queries

# ATR and Resolver Behaviour



# Measuring within the DNS




Query 1: a.b.example.com? to ns.example.com
Answer 1: NS nsb.z.example.com

*<discover name servers for z.example.com>*

Query 2: nsb.z.example.com to z.example.com
Answer 2: 192.0.2.1

Query 3 depends on the resolver successfully receiving answer 2

Query 3: a.b.example.com to 192.0.2.1
Answer 3: 10.0.0.1

# Experiment Details

- Use 6 tests:
  - 2 tests use ATR responses – one is DNS over IPv4, the other is DNS over IPv6
  - 2 tests use only truncated responses – IPv4 and IPv6
  - 2 tests use large fragmented UDP responses  - IPv4 and IPv6
- Use a technique of delegation without glue records (glueless) to perform the measurement entirely within the DNS
- Performed 55M experiments

# Looking at Resolvers

We are looking at resolvers who were passed “Answer 2” to see if they queried “Query 3”

| Protocol | Resolvers | ATR | Large UDP | TCP |
|---|---|---|---|---|
| IPv4 | 113,087 | 71.2% | 60.1% | 79.4% |
| IPv6 | 20,878 | 55.4% | 50.0% | 55.1% |

# Looking at Resolvers

We are looking at resolvers who were passed “Answer 2” to see if they queried “Query 3”

Inversely, lets report on the FAILURE rate of resolvers

| Protocol | Resolvers | Fail ATR | Fail Large UDP | Fail TCP |
|---|---|---|---|---|
| IPv4 | 113,087 | 28.8% | 39.9% | 20.6% |
| IPv6 | 20,878 | 44.6% | 50.0% | 44.9% |

# Seriously?

- More than one third of the ”visible” IPv4 resolvers are incapable of receiving a large fragmented packet
- And one half of the ”visible” IPv6 resolvers are incapable of receiving a large fragmented packet

# ASNs of **IPv4** Resolvers that do not followup when given a **large** UDP Response – Top 10

| ASN | Use | Exp | AS Name | CC |
|---|---|---|---|---|
| AS9644 | 0.78% | 447,019 | SK Telecom | KR |
| AS701 | 0.70% | 400,798 | UUNET - MCI Communications Services | US |
| AS17853 | 0.62% | 357,335 | LGTELECOM | KR |
| AS4766 | 0.59% | 340,334 | Korea Telecom | KR |
| AS4134 | 0.47% | 267,995 | CHINANET-BACKBONE | CN |
| AS31034 | 0.47% | 267,478 | ARUBA-ASN | IT |
| AS3786 | 0.39% | 225,296 | DACOM Corporation | KR |
| AS36692 | 0.38% | 217,306 | OPENDNS - OpenDNS | US |
| AS3215 | 0.33% | 189,810 | Orange | FR |
| AS812 | 0.30% | 169,699 | ROGERS COMMUNICATIONS | CA |

# ASNs of **IPv6** Resolvers that do not followup when given a **large** UDP Response – Top 10

| ASN | Use | Exp | AS Name | CC |
|---|---|---|---|---|
| AS15169 | 40.60% | 10,006,596 | Google | US |
| AS5650 | 0.90% | 221,493 | Frontier Communications | US |
| AS36692 | 0.84% | 206,143 | OpenDNS | US |
| AS812 | 0.78% | 193,073 | Rogers Communications Canada | CA |
| AS20057 | 0.46% | 114,440 | AT&T Mobility LLC | US |
| AS3352 | 0.38% | 92,925 | TELEFONICA_DE_ESPANA | ES |
| AS852 | 0.35% | 85,043 | TELUS Communications Inc. | CA |
| AS55644 | 0.32% | 80,032 | Idea Cellular Limited | IN |
| AS3320 | 0.25% | 61,938 | DTAG Internet service provider operations | DE |
| AS4761 | 0.24% | 60,019 | INDOSAT-INP-AP INDOSAT Internet Network Provider | ID |

# ASNs of **IPv4** Resolvers that do not followup in TCP when given a **truncated** UDP Response – Top 10

| ASN | Use | Exp | AS Name | CC |
|---|---|---|---|---|
| AS9299 | 0.55% | 252,653 | Philippine Long Distance Telephone | PH |
| AS24560 | 0.34% | 155,908 | Bharti Airtel | IN |
| AS3352 | 0.29% | 132,924 | TELEFONICA_DE_ESPANA | ES |
| AS9498 | 0.19% | 84,754 | BHARTI Airtel | IN |
| AS9121 | 0.14% | 61,879 | TTNET | TR |
| AS23944 | 0.13% | 58,102 | SKYBroadband | PH |
| AS9644 | 0.11% | 51,750 | SK Telecom | KR |
| AS24499 | 0.11% | 51,108 | Telenor Pakistan | PK |
| AS3215 | 0.10% | 43,614 | Orange | FR |
| AS23700 | 0.09% | 39,697 | Fastnet | ID |

# ASNs of **IPv6** Resolvers that do not followup in TCP when given a **truncated** UDP Response – Top 10

| ASN | Use | Exp | AS Name | CC | |
|---|---|---|---|---|---|
| AS15169 | 4.15% | 961,287 | Google | US | |
| AS21928 | 1.72% | 399,129 | T-Mobile USA | US | |
| AS7922 | 1.57% | 364,596 | Comcast Cable | US | |
| AS3352 | 0.54% | 126,146 | TELEFONICA_DE_ESPANA | ES | |
| AS22773 | 0.38% | 87,723 | Cox Communications Inc. | US | |
| AS55644 | 0.35% | 80,844 | Idea Cellular Limited | IN | |
| AS20115 | 0.31% | 71,831 | Charter Communications | US | |
| AS20057 | 0.30% | 70,518 | AT&T Mobility | US | |
| AS6713 | 0.20% | 46,196 | IAM-AS | MA | |
| AS8151 | 0.20% | 45,754 | Uninet S.A. de C.V. | MX | |

# What's the impact?

- Failure in the DNS is often masked by having multiple resolvers in the clients local configuration
- And the distribution of users to visible recursive resolvers is heavily skewed (10,000 resolvers by IP address handle the DNSqueries of more than 90% of end users)
- So to assess the impact lets look at the results by counting user level success / failure to resolve these glueless names

# Looking at Users

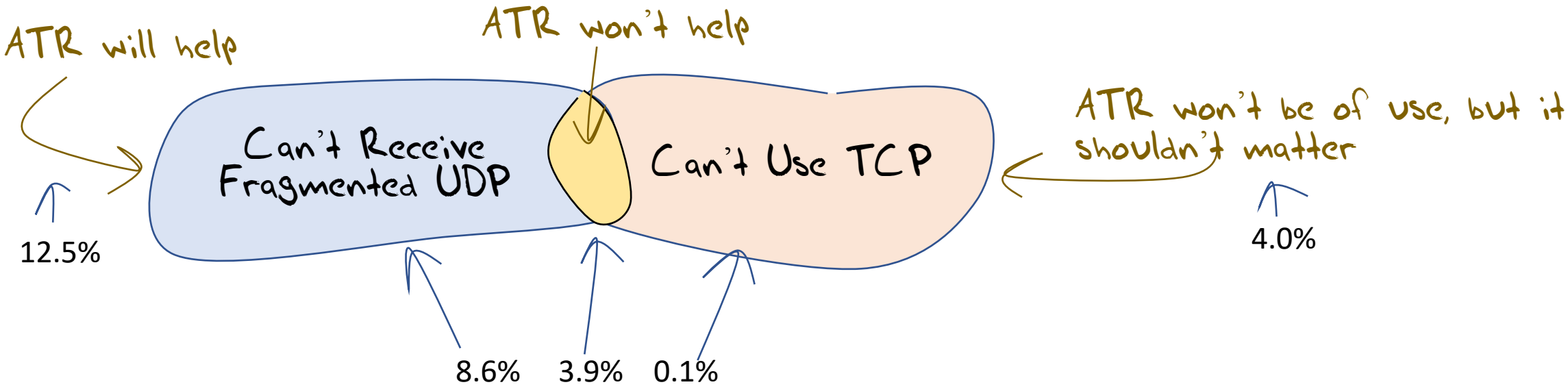
- Rather than looking at individual resolvers being able to pose Question 3, lets count:
  - A “success” if any resolver can query Question 3 on behalf of the user
  - A “failure” is recorded when no resolver generates a query to Question 3

# Looking at Users - Failure Rates

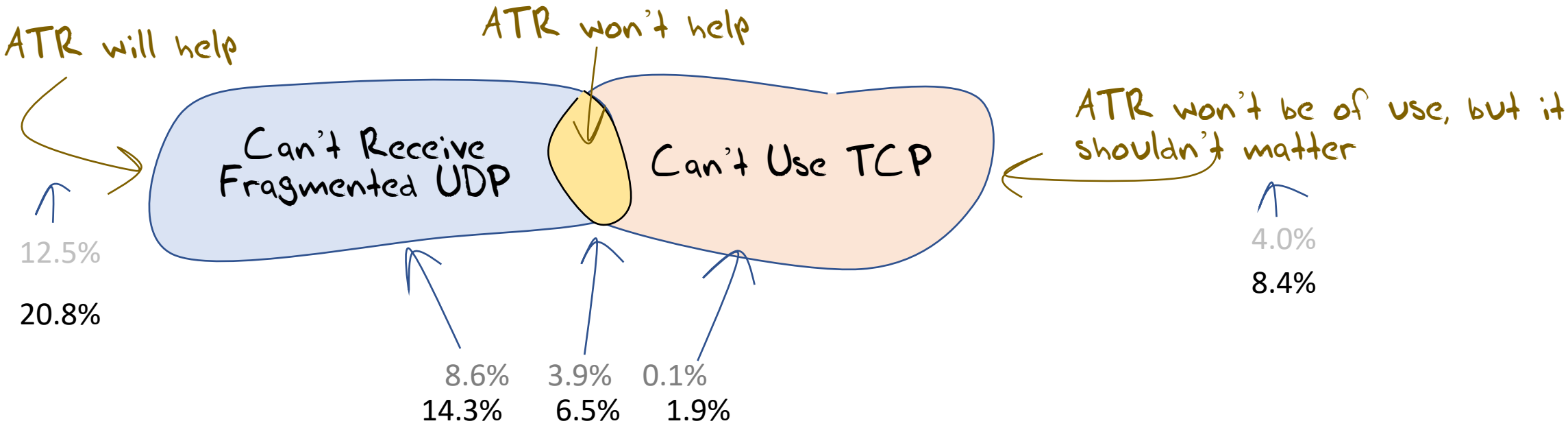
| IPv4 | | IPv6 | |
|---|---|---|---|
| UDP Frag: | 12.5% | UDP Frag: | 20.8% |
| TCP: | 4.0% | TCP: | 8.4% |
| ATR | 3.9% | ATR | 6.5% |

These loss rates are expressed as an estimated percentage of users,

# ATR and Resolver Behaviour – IPv4



# ATR and Resolver Behaviour – IPv4 IPv6



# Net Change in User Failure Rates

| **IPv4** | | **IPv6** | |
|---|---|---|---|
| Fragged UDP Loss: | 12.5% | Fragged UDP Loss: | 20.5% |
| ATR Loss Rate: | 3.9% | ATR Loss Rate: | 6.5% |

# ATR Assessment

- Is this level of benefit worth the additional server and traffic load when sending large responses?
- Is this load smaller than resolver hunting in response to packet drop?
- It the faster fallback to TCP for large responses a significant benefit?
- Is 10ms ATR delay too short? Would a longer gap reduce response reordering? Do we care?
- Do we have any better ideas about how to cope with large responses in the DNS?

Thanks!