Is the DNS ready for IPv6?

Geoff Huston AM
APNIC Labs
DNS OARC 42, 9 Feb 2024

IETF Best Current Practice – BCP 91

RFC3901 – September 2004 “DNS IPv6 Transport Operational Guidelines”:

- Every recursive name server SHOULD be either IPv4-only or dual stack
- Every DNS zone SHOULD be served by at least one IPv4-reachable name server

Which is saying as an IPv6 Operational guideline “you better keep IPv4 going”

The RFC actually says very little about IPv6!

Proposed: 3901bis

Current IETF draft proposed to update RFC3901 by saying:

- It is RECOMMENDED that are least two NS for a zone are dual stack name servers
- Every authoritative DNS zone SHOULD be served by at least one IPv6-reachable authoritative name server

Which is saying as an IPv6 Operational guideline "time to take IPv6 seriously" and NOT saying that servers need to keep IPv4 around – which is largely the opposite of the advice in RFC 3901!

The assumption behind 3901bis

- That IPv6 is now a mature and well understood technology, and using IPv6 as the transport for the DNS is as efficient and as fast as using IPv4

The assumption behind 3901bis

- That IPv6 is now a mature and technology, and using IPv6 as the transport efficient and as fast as using IPv4

Is this really true?

IPv6 and the DNS

How well is IPv6 supported in the DNS?

1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a “happy eyeballs” version of DNS server selection?
   - Or is there a reverse bias to use IPv4?
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you?
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

A word or two about "how" to talk about the DNS

We really don’t understand what a “resolver” is!

- It could be a single platform running an instance of DNS resolver code
- It could be a collection of independent back-end systems with a load distributor front end facing clients
- It could be a hybrid collection where the back ends synchronise each other to emulate a common cache
- It is a stub, recursive, or forwarding resolver
- A resolver may have 1 client or millions of clients or anything in between

When we talk about “resolvers” it’s challenging to understand exactly what we are talking about!

Another word, this time about "how" to talk about DNS queries

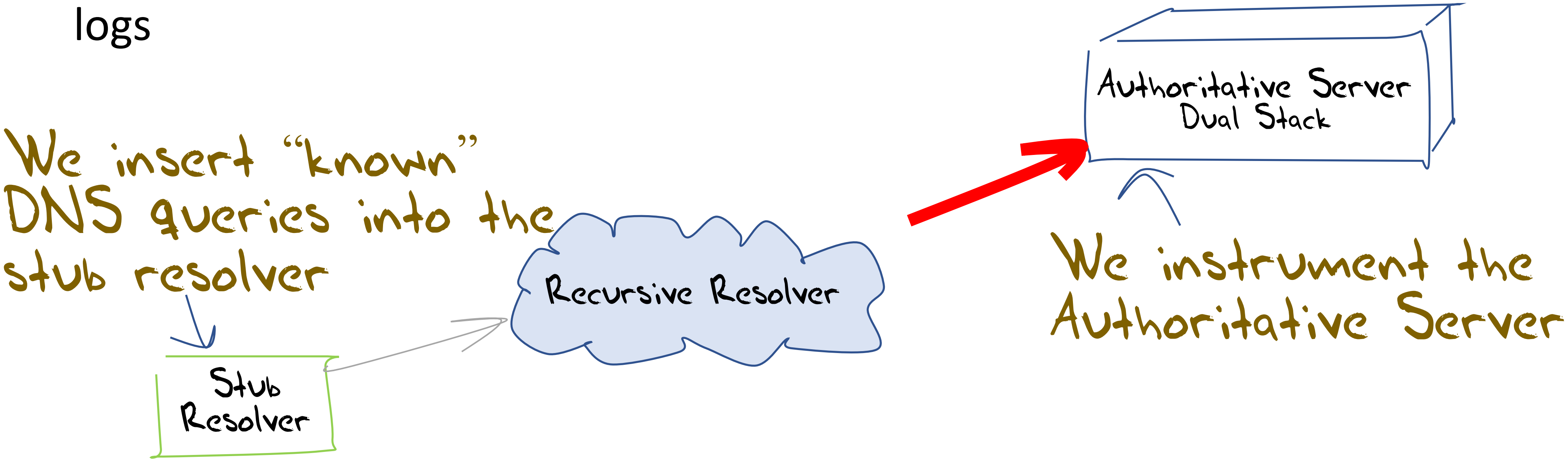
We don’t understand what a query is!

- Which sounds silly, but the distributed resolution process causes a ‘fan out’ of queries as part of the resolution process when a single query may cause a number of ‘discovery’ queries to establish the identity of the authoritative server(s) for the name
- Resolvers all use their own timers for retransmission
- Queries have no “hop count” or “resolver path” attached
  - there is no context to understand the reason for a query!
- Queries have a life of their own

APNIC's DNS Experimental Rig

- We use the ad network to “seed” DNS queries
- We make parts of the name unique to each measurement

  That way the recursive resolvers have no cached data and are forced to query the authoritative server
- We observe the recursive to authoritative query process by instrumenting the authoritative server, and match experiment placement records to the server’s DNS logs



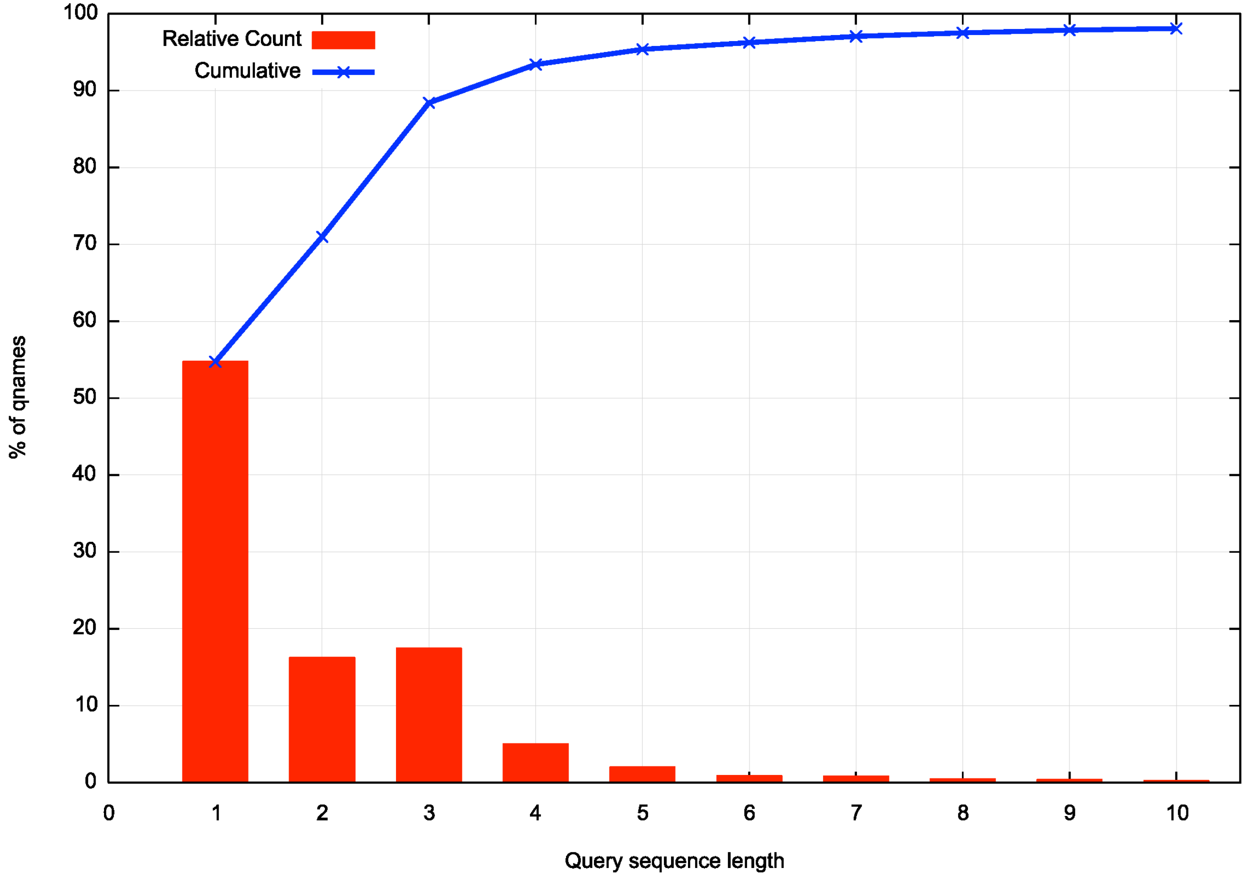
We insert “known” DNS queries into the stub resolver

We instrument the Authoritative Server

Also, the DNS is VERY noisy!

There are a lot of gratuitous DNS queries

- Some 46% of qnames are queried 2 or more times
- Some 30% are queried 3 or more times!



Dual Stack DNS

How well is IPv6 supported in the DNS?

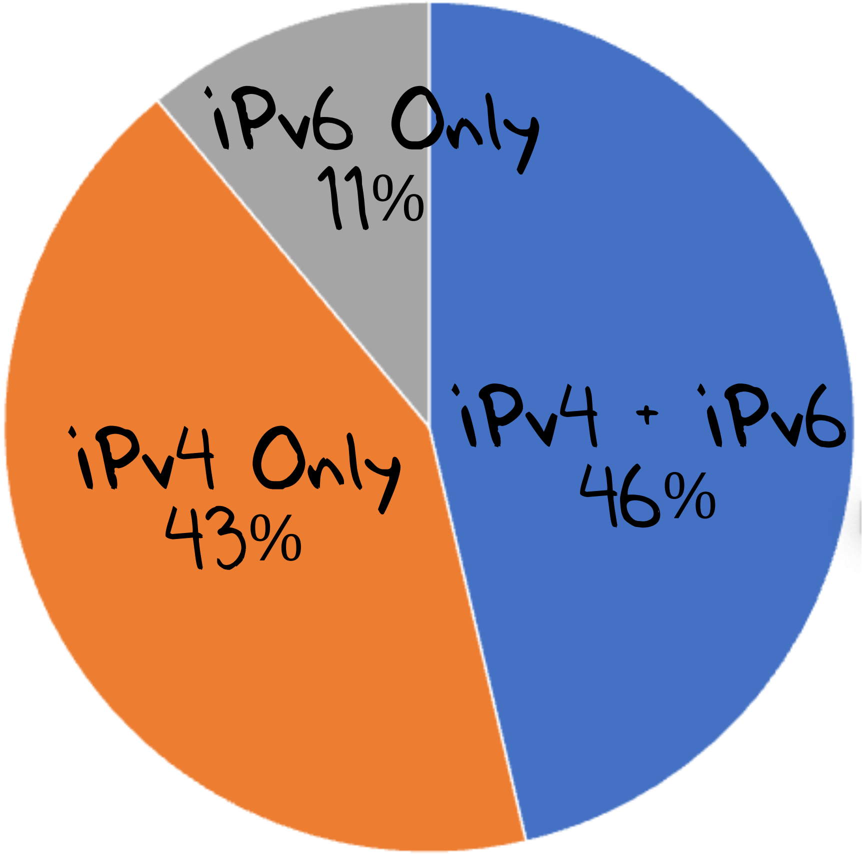
1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a “happy eyeballs” version of DNS server selection?
   - Or is there a reverse bias to use IPv4?
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you?
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

Dual Stack DNS

A "happy eyeballs*" DNS approach would be to prefer to use the IPv6 address of the authoritative server in preference to the IPv4 address

A "reverse bias" DNS approach would be to prefer to use the IPv4 address

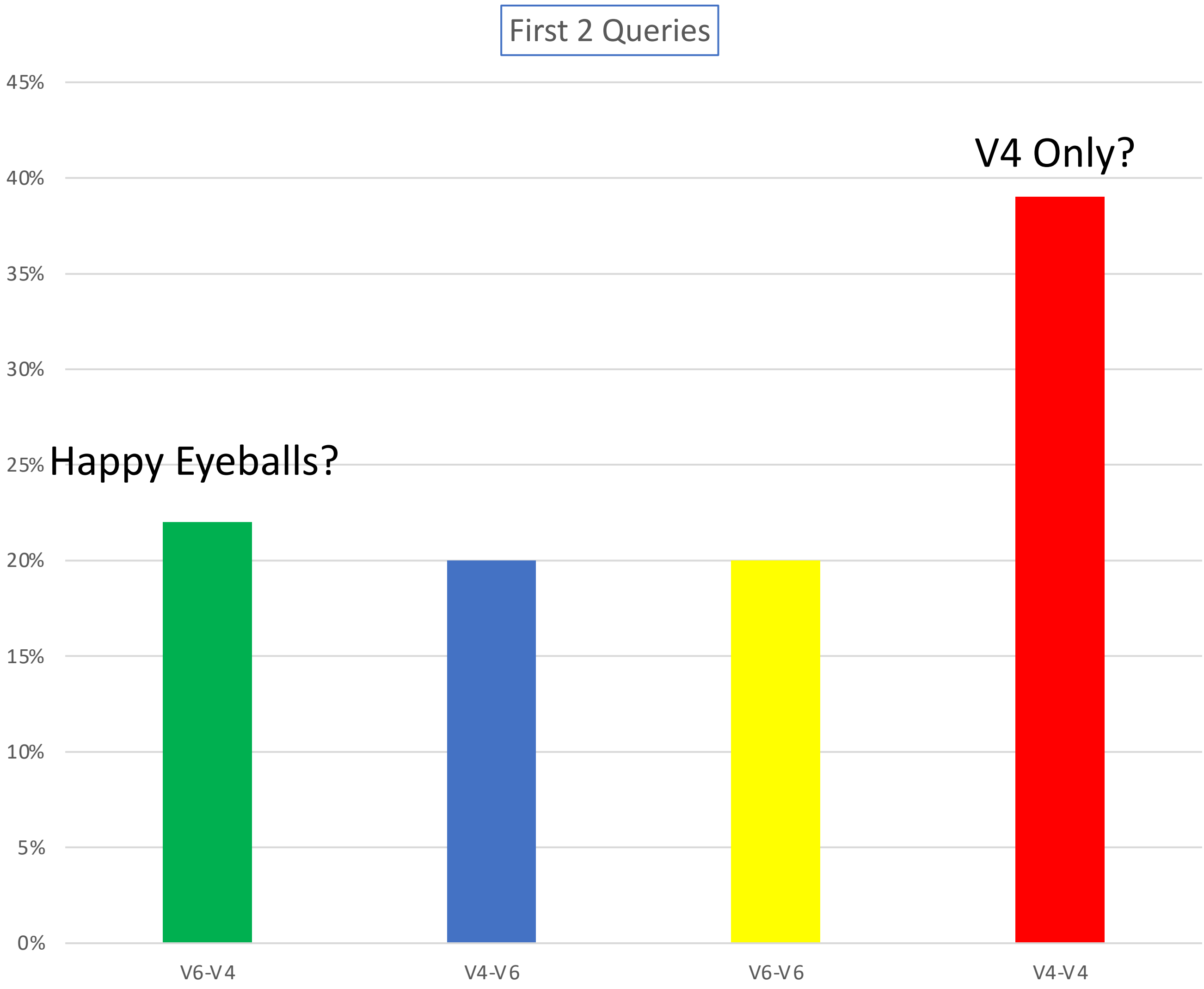
Data collected Dec 23 – Jan 24 using 445M domain names



* Why is happy eyeballs important?

Dual Stack DNS

A “happy eyeballs” DNS approach would be to prefer to use the IPv6 address of the authoritative server in preference to the IPv4 address and follow this initial query with a IPv4 query soon after

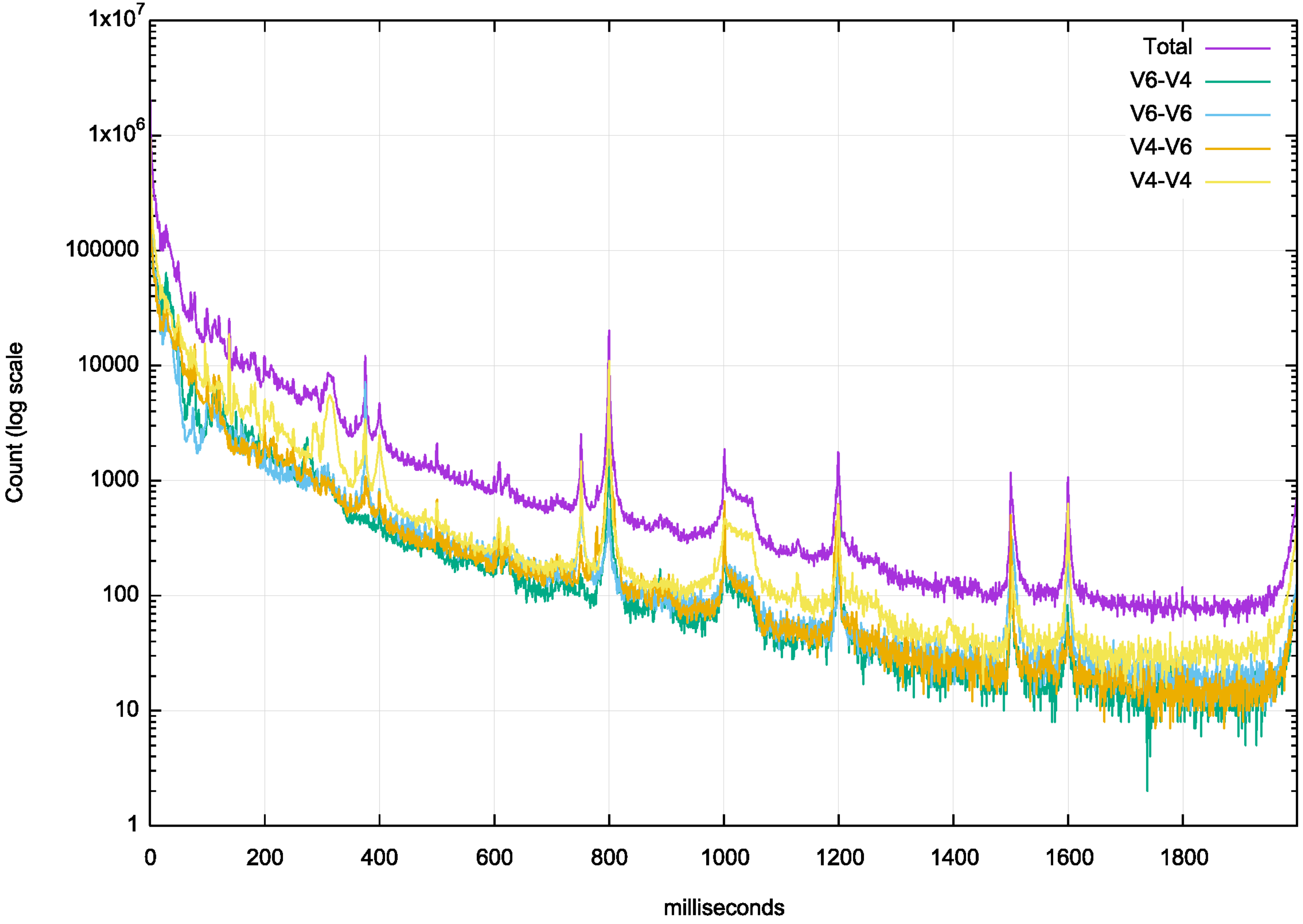


Dual Stack DNS

A “happy eyeballs” DNS approach would be minimise the delay between the initial 2 queries

But what we see is evidence of conventional DNS timeout values of 370ms, 400ms, 800ms and 1 sec

Delay between first 2 Queries



Dual Stack DNS

How well is IPv6 supported in the DNS?

1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a "happy eyeballs" version of DNS server selection? No!
   - Or is there a reverse bias to use IPv4? Yes!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you?
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

Dual Stack DNS

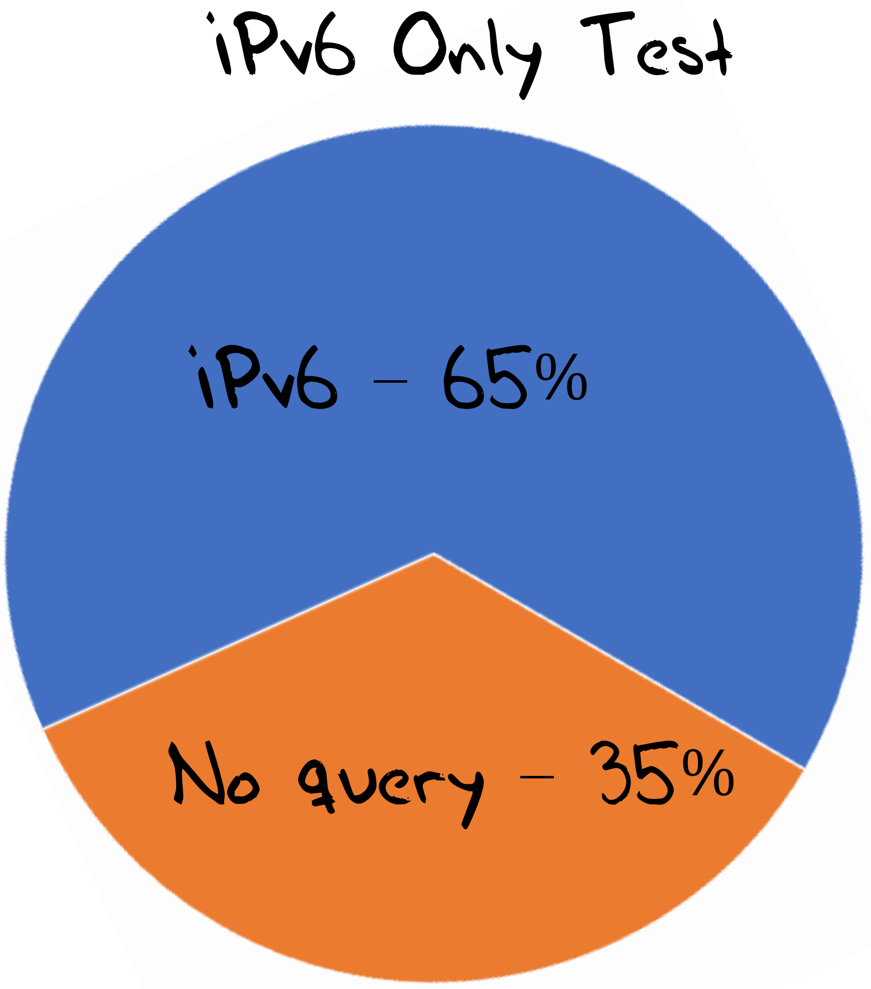
How well is IPv6 supported in the DNS?

1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a "happy eyeballs" version of DNS server selection? No!
   - Or is there a reverse bias to use IPv4? Yes!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you?
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

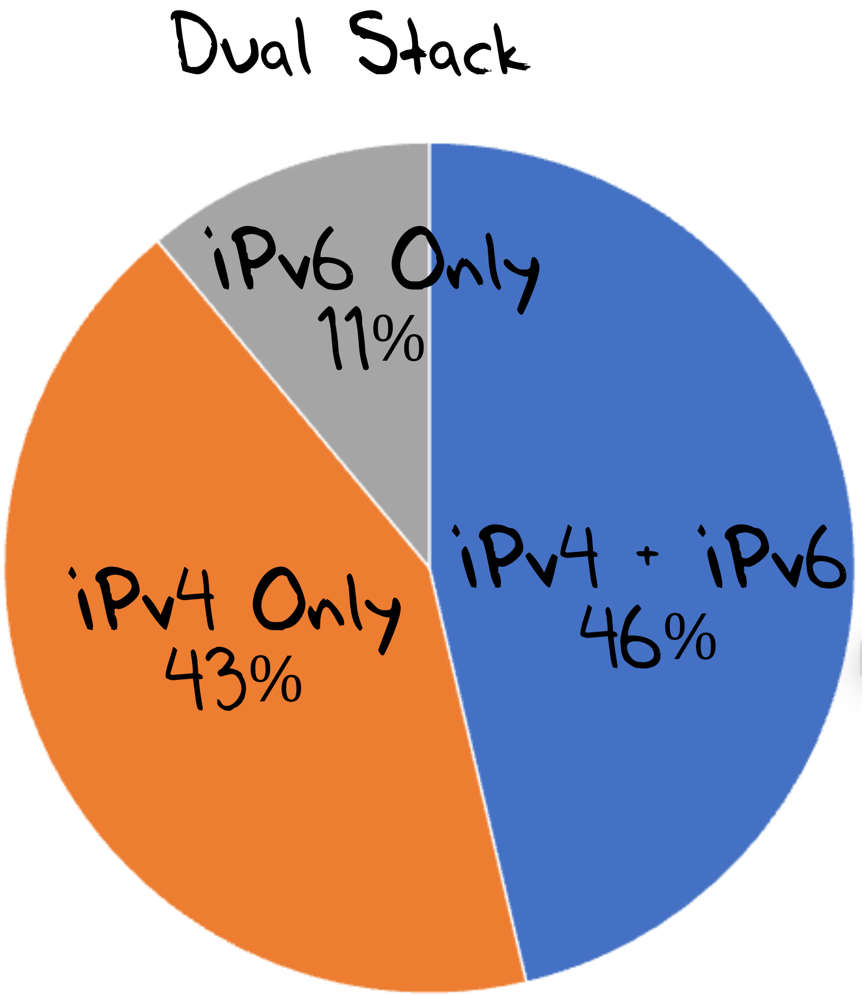
Dual Stack vs IPv6 only DNS

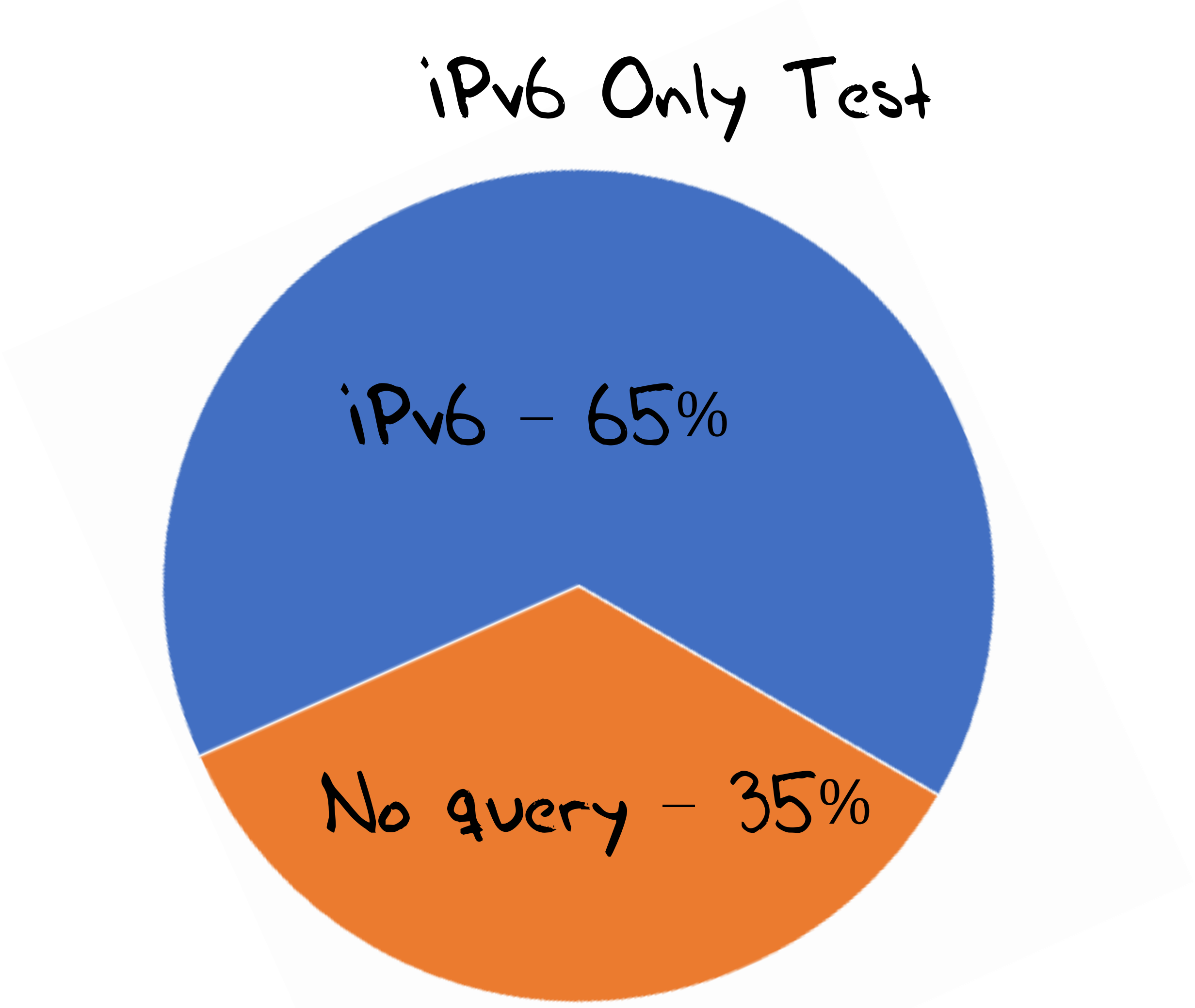
In this case the authoritative name server only has an IPv6 address

Of all the clients that are presented with an experiment (51M over 5 days) 65% of names are seen asking for the experiment name if the DNS server is reachable over IPv6 only



Dual Stack vs IPv6 only DNS





Dual Stack DNS

How well is IPv6 supported in the DNS?

1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a "happy eyeballs" version of DNS server selection? No!
   - Or is there a reverse bias to use IPv4? Yes!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you? 65%
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

Dual Stack DNS

How well is IPv6 supported in the DNS?

1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a "happy eyeballs" version of DNS server selection? No!
   - Or is there a reverse bias to use IPv4? Yes!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you? 55%
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

IPv6 and Packet Fragmentation

IPv6 made two major changes to IP’s handling of packet fragmentation:

- The fragmentation control header has been moved out of the IP header to become an **extension header**
  - In other words, the UDP / TCP protocol header is pushed further into the packet and to find it you need to follow the header chain
- The IPv4 ‘Don’t Fragment’ bit is jammed **on** in IPv6
  - In the case of path MTU issues IPv6 routers should not perform fragmentation on the fly, but are required to pass an ICMPv6 PTB message back to the packet’s sender

Who uses Fragmentation anyway?

- Well, the DNS is a good place to start looking!

Who uses large DNS packets anyway?

- And the root zone DNS is a good place to start looking!

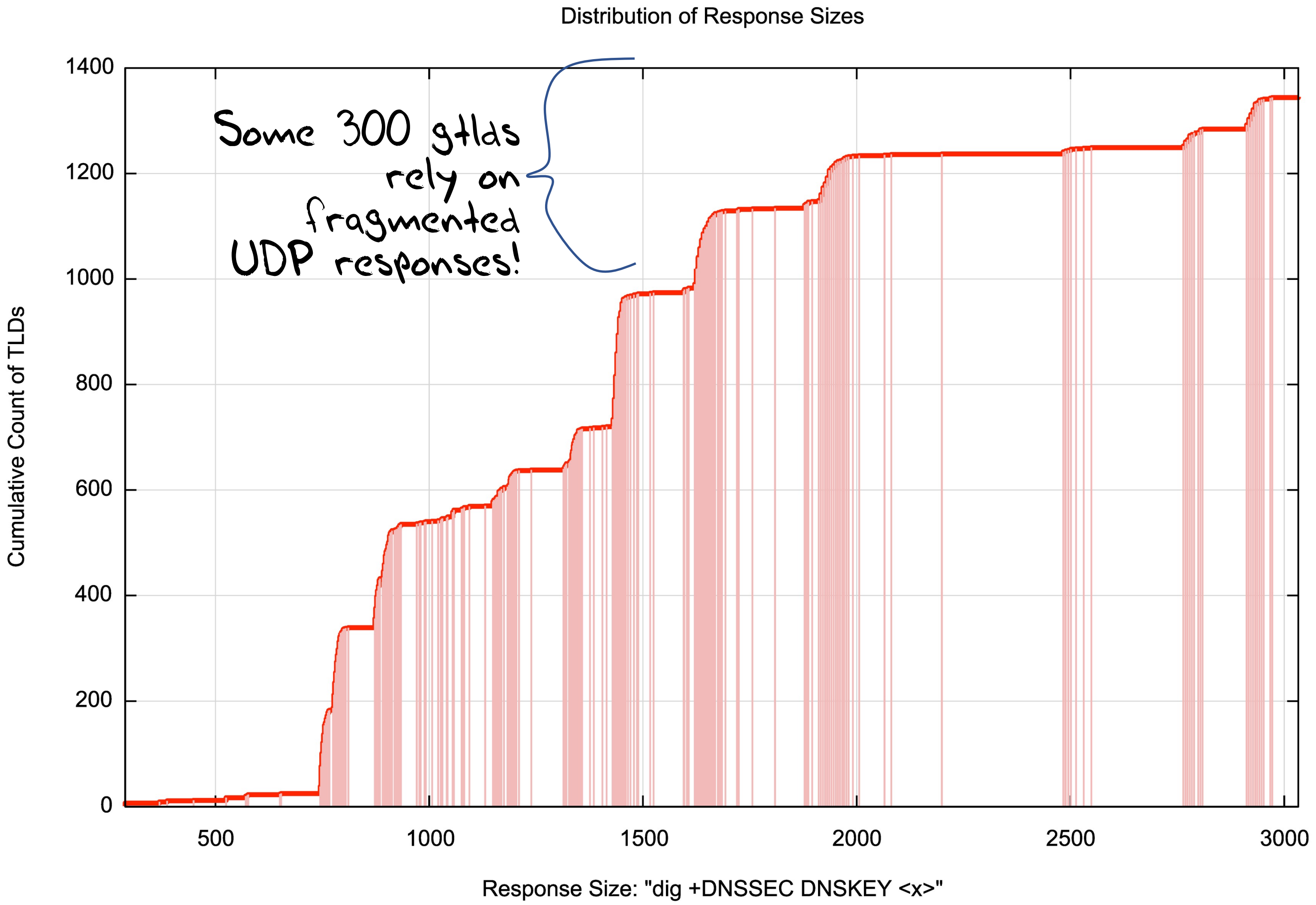
Who uses large DNS packets anyway?

These folk do!

```
.sl 3319
.pl 2193
.gdn 1954
.ve 1951
.uy 1951
.bg 1951
.xn--mgbx4cd0ab 1931
.africa 1897
.ad 1769
.ss 1715
.firmdale 1693
.xn--mgbah1a3hjkrd 1691
.xn--mgbt3dhd 1681
.ar 1675
.nowruz 1669
.beats 1667
.apple 1667
.shia 1665
.pars 1665
.tci 1663
.zm 1661
.td 1661
.si 1661
.na 1661
.ly 1661
.kw 1661
.ke 1661
.gy 1661
.lifestyle 1638
.living 1629
```

Size of dnssec-signed DNSKEY response for some gtlds in Nov-23

Who uses large DNS packets anyway?



However…

UDP Fragmentation has its problems

- UDP trailing fragments in IPv4 and IPv6 may encounter fragment filtering rules on firewalls in front of resolvers

- Large UDP packets in IPv6 may encounter path MTU mismatch problems, and the ICMP6 Packet Too Big diagnostic message may be filtered.

  Even if it is delivered, the host may not process the message due to the lack of verification of the authenticity of the ICMP6 message.

  Because the protocol is UDP, receipt of an ICMP6 message will not cause retransmission of a re-framed packet.

- UDP fragments in IPv6 are implemented by Extension Headers. There is ample evidence of deployment of IPv6 switching equipment that unilaterally discards IPv6 packets with extension headers

Is this a problem for today's IPv6 Internet?

- Can we measure the extent to which users might be affected with this scenario of large DNS responses, DNS resolvers and IPv6?

Our Measurement Approach

We use an Online Ad platform to enroll endpoints to attempt to resolve a set of DNS names:

- Each endpoint is provided with a unique name string (to eliminate the effects of DNS caching)
- The DNS name is served from our authoritative servers
- Resolving the DNS name requires the user's DNS resolvers to receive a fragmented IPv6 packet

V6, the DNS and Fragmented UDP

Total number of tests (DNS over UDP over IPv6): 32,951,595

Failure Rate in receiving a large response: 18,557,838

IPv6 Fragmentation Failure Rate: **56%**

Data gathered 20 Dec 2023 – 9 Jan 2024

V6, the DNS and Fragmented UDP

Total number of tests (DNS over UDP over IPv6): 32,951,595

Failure Rate in receiving a large response: 18,557,838

IPv6 Fragmentation Failure Rate: **56%**

That's awesomely bad!

Data gathered 20 Dec 2023 – 9 Jan 2024

What to do?

Accepting a future IPv6-only Internet means we are going to have to take the problem of IPv6 Fragmentation seriously

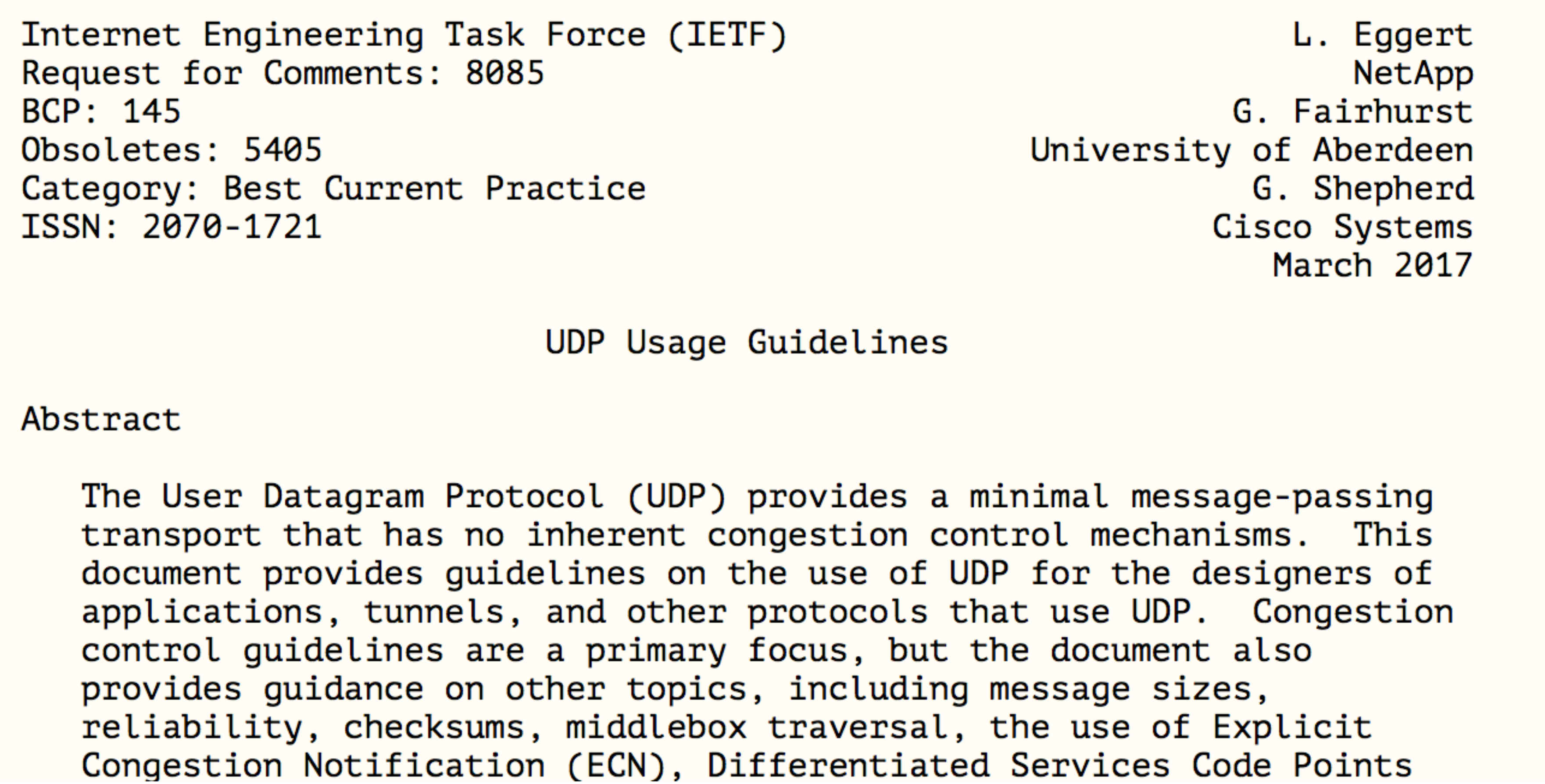
- Because relying on IPv4 as a backup is a hack with an indeterminate future!

Which means that we need to figure out how to change the appalling drop rate for fragmented IPv6 packets both in the DNS and in end-to-end paths

Should we try and fix the network problem or try to work around it?

What do the RFC's say?

What do the RFC's say?



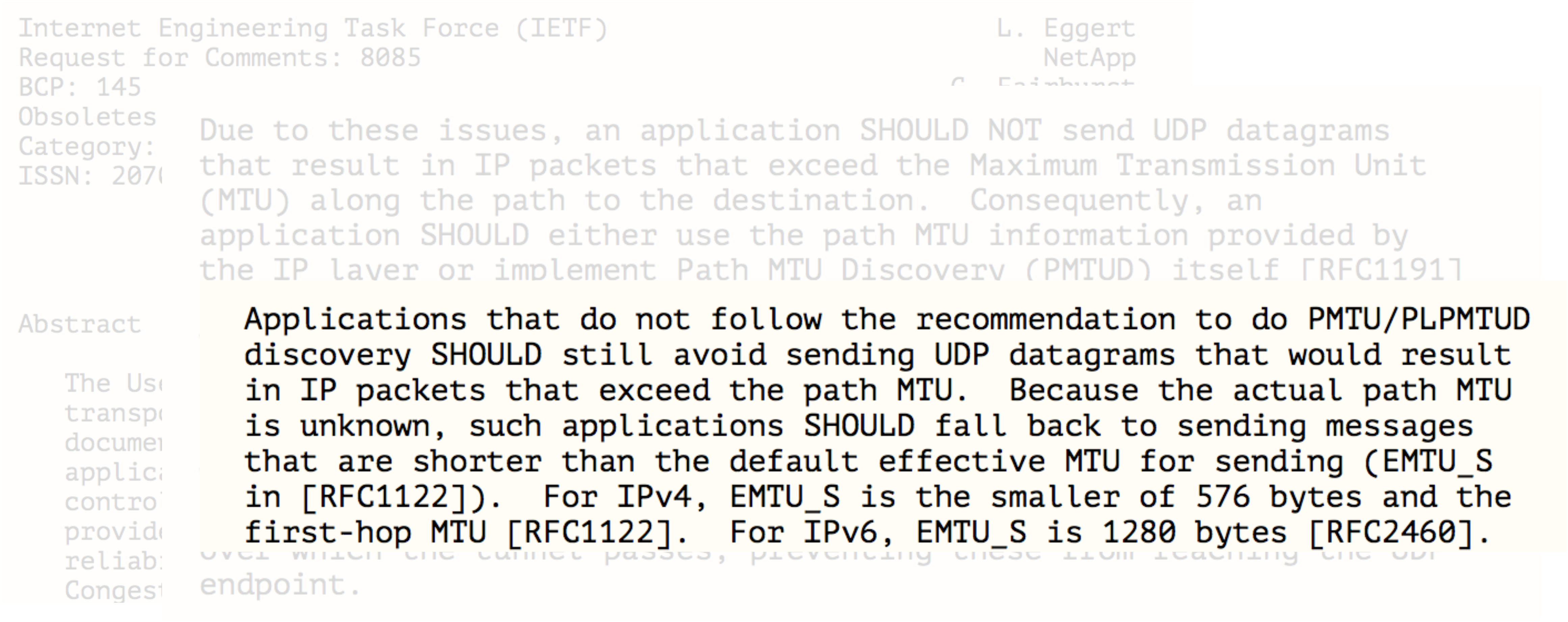
Internet Engineering Task Force (IETF) L. Eggert
Request for Comments: 8085 NetApp
BCP: 145 G. Fairhurst
Obsoletes: 5405 University of Aberdeen
Category: Best Current Practice G. Shepherd
ISSN: 2070-1721 Cisco Systems
March 2017

UDP Usage Guidelines

Abstract

The User Datagram Protocol (UDP) provides a minimal message-passing transport that has no inherent congestion control mechanisms. This document provides guidelines on the use of UDP for the designers of applications, tunnels, and other protocols that use UDP. Congestion control guidelines are a primary focus, but the document also provides guidance on other topics, including message sizes, reliability, checksums, middlebox traversal, the use of Explicit Congestion Notification (ECN), Differentiated Services Code Points

What do the RFC's say?



Internet Engineering Task Force (IETF) L. Eggert
Request for Comments: 8085 NetApp
BCP: 145
Obsoletes
Category:
ISSN: 207

Abstract

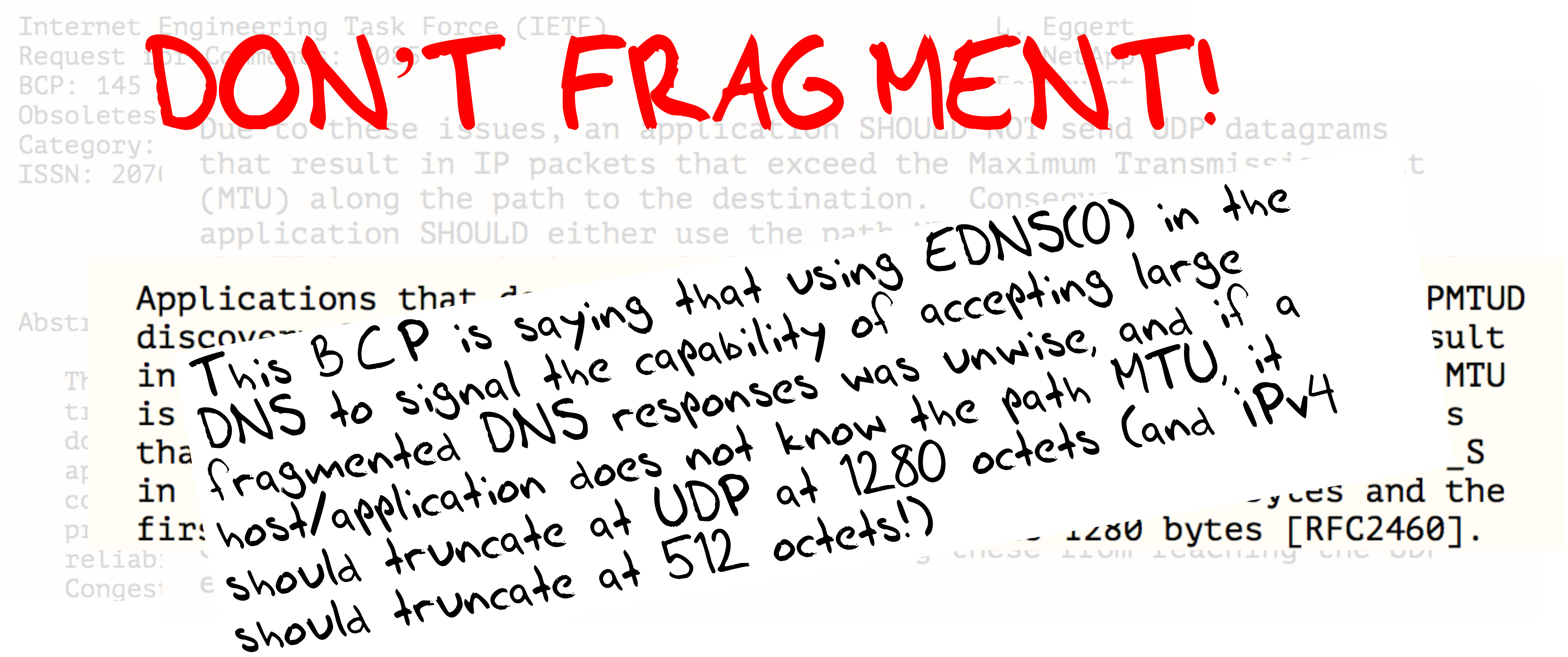
The Us
transp
docume
applica
contro
provid
reliab
Conges

Due to these issues, an application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself [RFC1191]

over which the tunnel passes, preventing these from reaching the UDP endpoint.

Applications that do not follow the recommendation to do PMTU/PLPMTUD discovery SHOULD still avoid sending UDP datagrams that would result in IP packets that exceed the path MTU. Because the actual path MTU is unknown, such applications SHOULD fall back to sending messages that are shorter than the default effective MTU for sending (EMTU_S in [RFC1122]). For IPv4, EMTU_S is the smaller of 576 bytes and the first-hop MTU [RFC1122]. For IPv6, EMTU_S is 1280 bytes [RFC2460].

What do the RFC's say?



```
Internet Engineering Task Force (IETF)                    L. Eggert
Request for Comments: 8085                                   NetApp
BCP: 145
Obsoletes:
Category:
ISSN: 207

Abstr

  Th
  tr
  do
  ap
  co
  pr
  reliab
  Congest
```

DON'T FRAGMENT!

```
Due to these issues, an application SHOULD NOT send UDP datagrams
that result in IP packets that exceed the Maximum Transmiss      t
(MTU) along the path to the destination.  Conseq
application SHOULD either use the path M
```

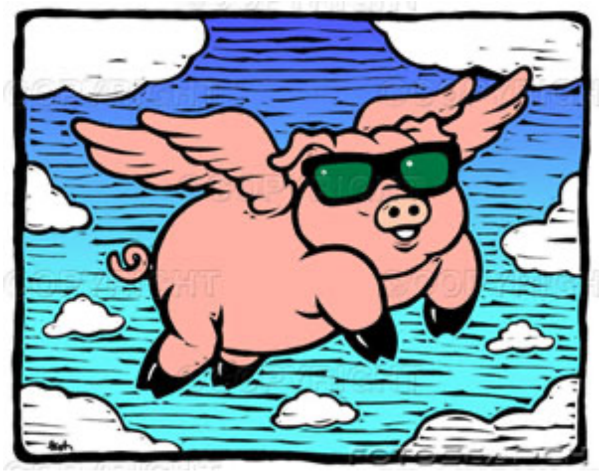
```
Applications that d                                        PMTUD
discover                                                    sult
in                                                           MTU
is                                                             s
tha                                                           _S
in                                                 ytes and the
firs                                  1280 bytes [RFC2460].
```

This BCP is saying that using EDNS(0) in the DNS to signal the capability of accepting large fragmented DNS responses was unwise, and if a host/application does not know the path MTU, it should truncate at UDP at 1280 octets (and IPv4 should truncate at 512 octets!)

What can we do about it?

Fix it!

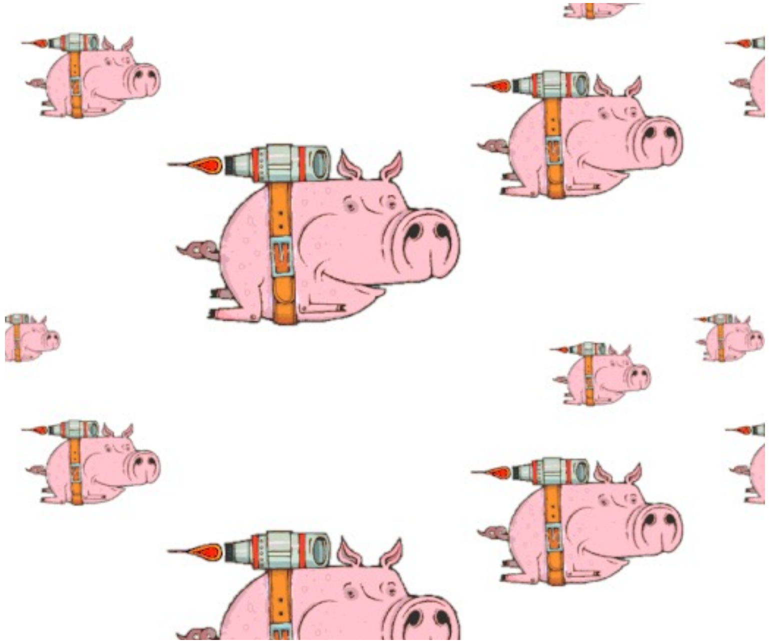
Get all the deployed routers, switches and firewalls and related network middleware to accept packets with IPv6 Fragmentation Headers



What can we do about it?

Change it!

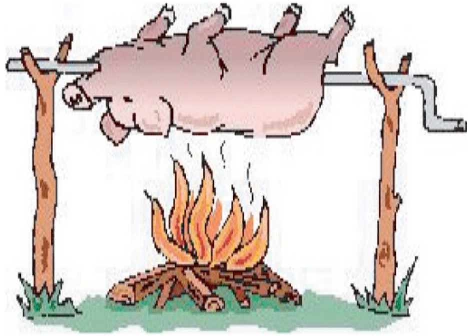
Change the way in which IPv6 manages IP fragmentation and the use of Extension Headers as Fragmentation Control fields



What can we do about it?

Avoid it!

Change application behaviour to avoid the use of packet fragmentation completely



Large DNS Responses and IPv6

Change the transport protocol?

- DNS over TCP by default
- DoT by default
- DoH
- DoQUIC

or

- Devise some new DNS framing protocol that uses multiple packets with firewall-friendly packet and protocol headers instead of IP fragmentation

Large DNS Responses and IPv6

Change the application protocol behaviour?

- Perform UDP MTU discovery using EDNS(0) UDP Buffer Size variations as a probe
- Shift Additional Records into additional explicit UDP query/response transactions rather than bloating the original DNS response
- Add a truncated minimal UDP response to trail a fragmented response (ATR)

Truncate and failover to TCP

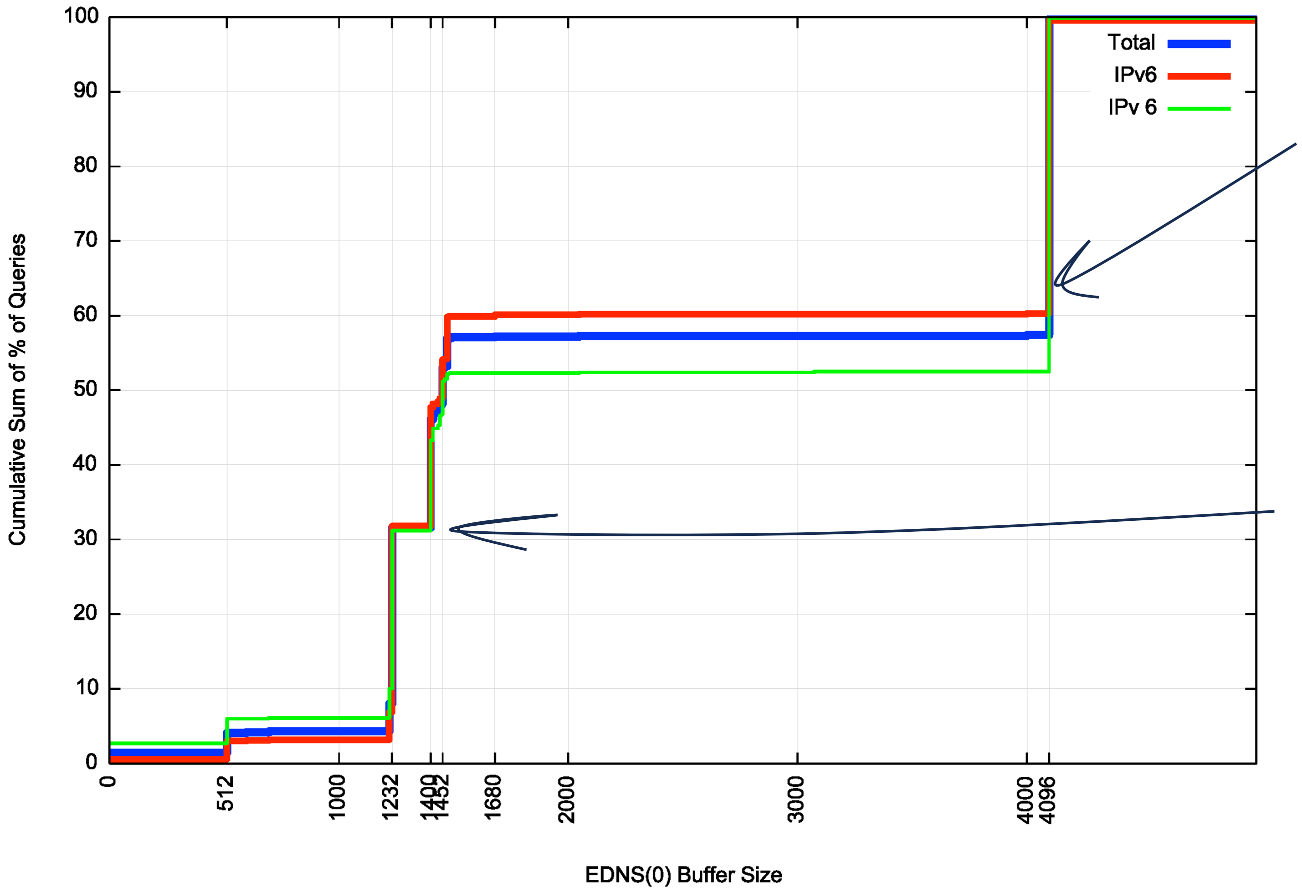
- Use an EDNS Buffer Size in queries to ensure that IPv6 responses are never fragmented
- Large responses will be truncated
- The truncation should trigger the querier to perform an immediate followup of the same query, using TCP

- Which means that we are probably looking at working around the problem by changing the configuration of DNS queries and use an EDNS buffer size of 1232 octets

See https://dnsflagday.net/2020/

DNS Flag Day 2020 Uptake

We know what we can do to make this work well, but we seem to be reluctant to actually do it!



46% of IPv6 queries are still using 4096 Byte Buffer Size

69% of IPv6 queries are using a Buffer Size greater than 1232

Is the DNS ready for IPv6?

- Not really!

Thanks!

Additional Commentary

DNS and UDP

- The original DNS spec uses USP with a maximum payload size of 512 octets (RFC 1035, sec 2.3.4)
- RFC2671 defined an Extension Mechanism that included a buffer size parameter to permit DNS payloads in UDP larger than 512 (sec 2.3)
- RFC 6891 proposes a buffer size of 4,096 as a “starting point” (sec 6.2.5) and proposes a fallback to a non-fragmented size before using truncation
- If the response cannot fit in the UDP payload the responder sets the Truncation bit in its response, which signals to the querier that it should retry using TCP

DNS UDP Responder

- If the response can fit in the EDNS Buffer size, then generate a UDP packet and let the network (IPv4) or host (IPv6) fragment the UDP packet as required
  - Any received ICMP message will NOT cause re-transmission!
- Otherwise set the TC bit

IPv6

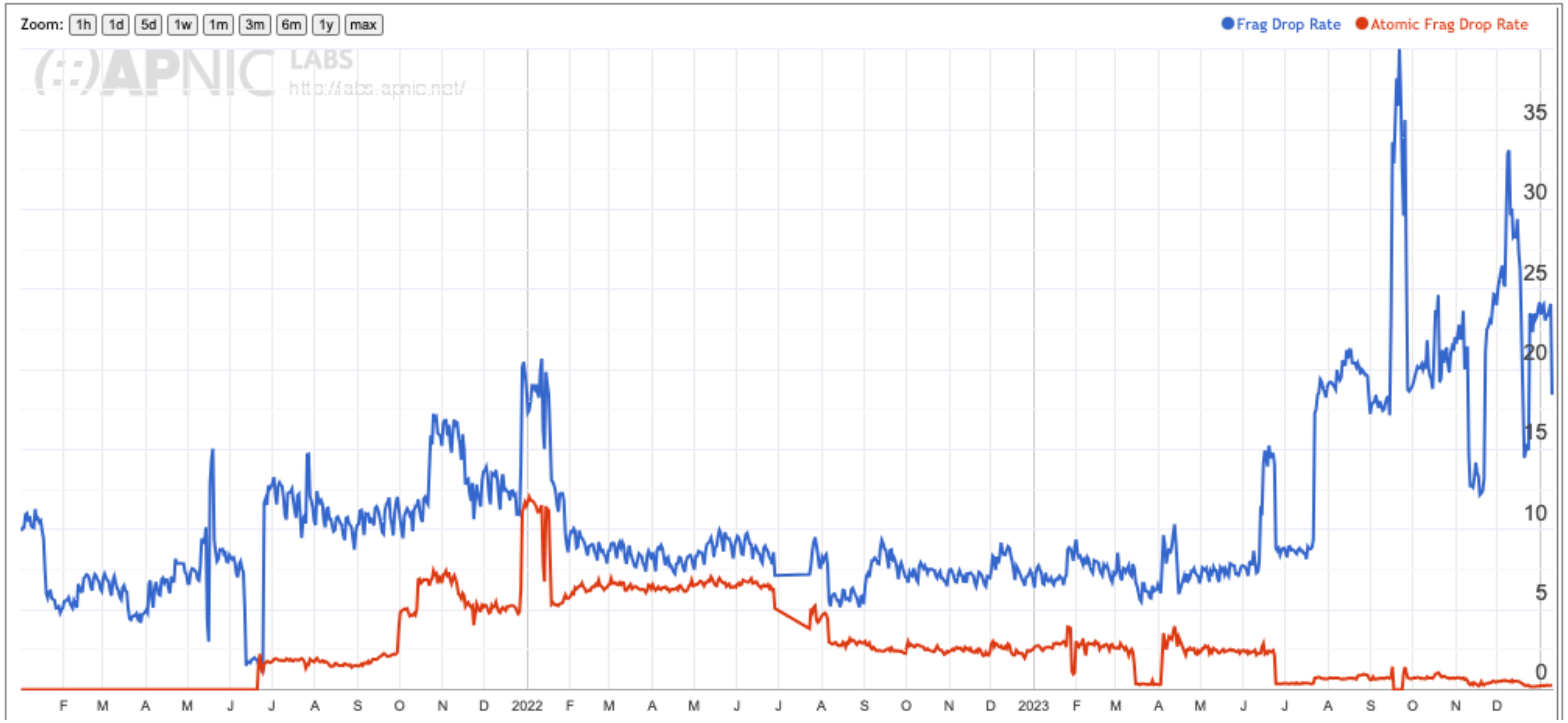
- IPv6 will not allow routers to *forward fragment*
- IPv6 relies on receiving a ICMP6 Packet Too Big message with a MTU size
  - The MTU value is locally cached for an interval, and used for…
- An IPv6 sender must perform outgoing packet fragmentation, using an inserted IPv6 Fragmentation Extension Header

IPv6 issues

- ICMP6 messages are often blocked
- ICMP6 messages cannot be validated
- Anycast may result in misdirected ICMP6 messages
- IPv6 Fragmented Packets are often dropped
- A lost response means that the querier has to timeout
  - DNS dimeout timers range from ~400ms to 1 second!

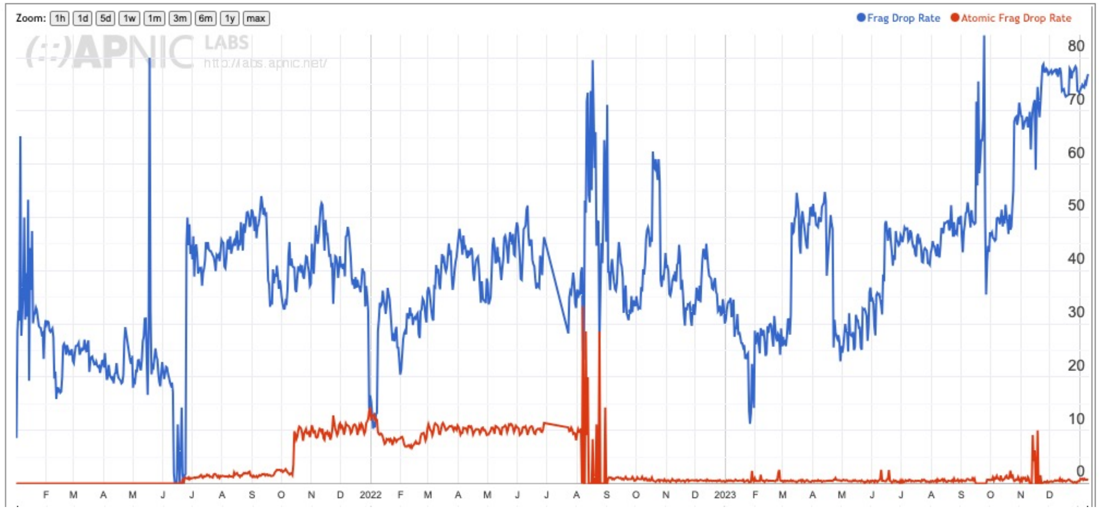
V6 Fragmentation Drop Rates

Use of V6FRAG Drop Rate for World (XA)

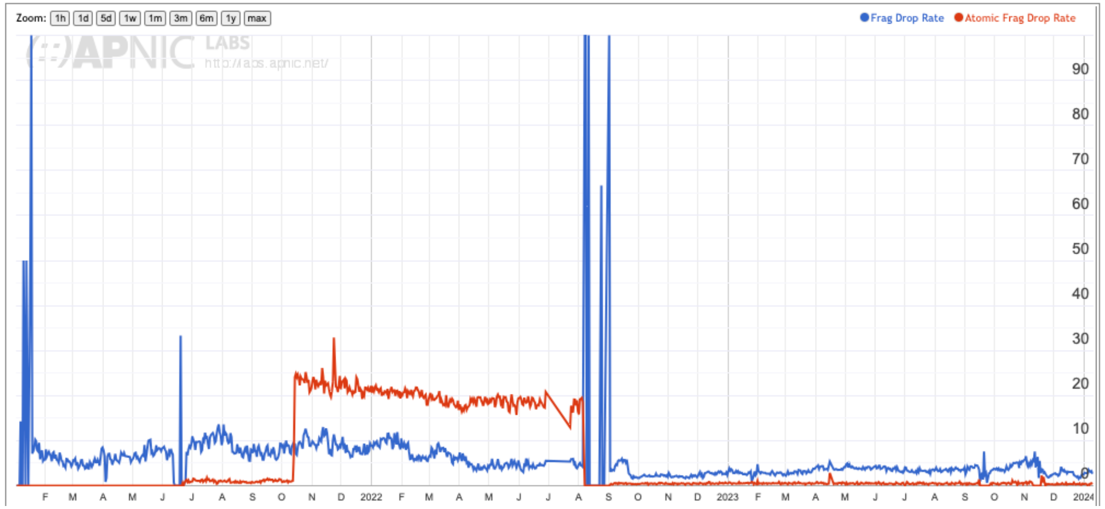


V6 Frag Drop is highly variable

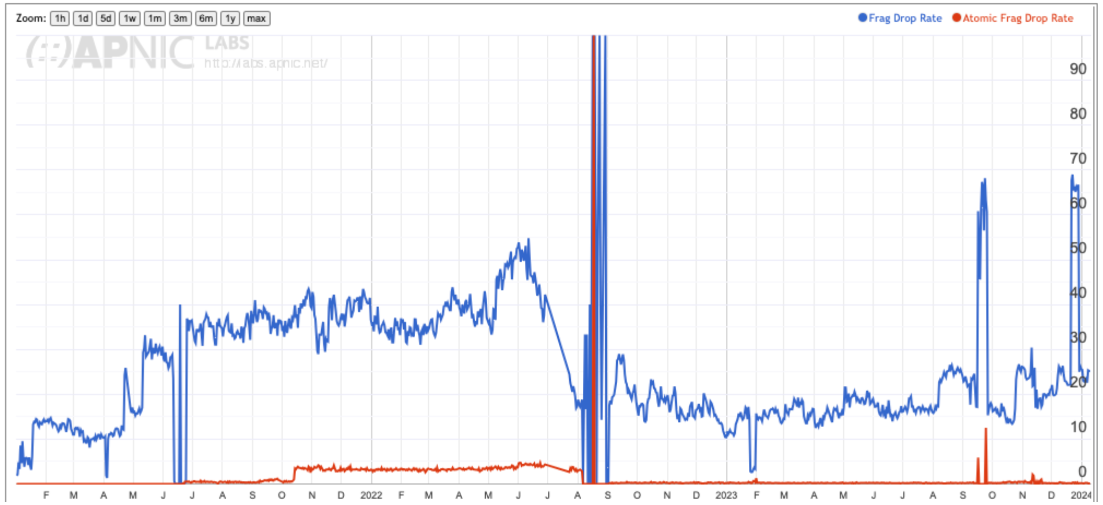
Use of V6FRAG Drop Rate for China (CN)



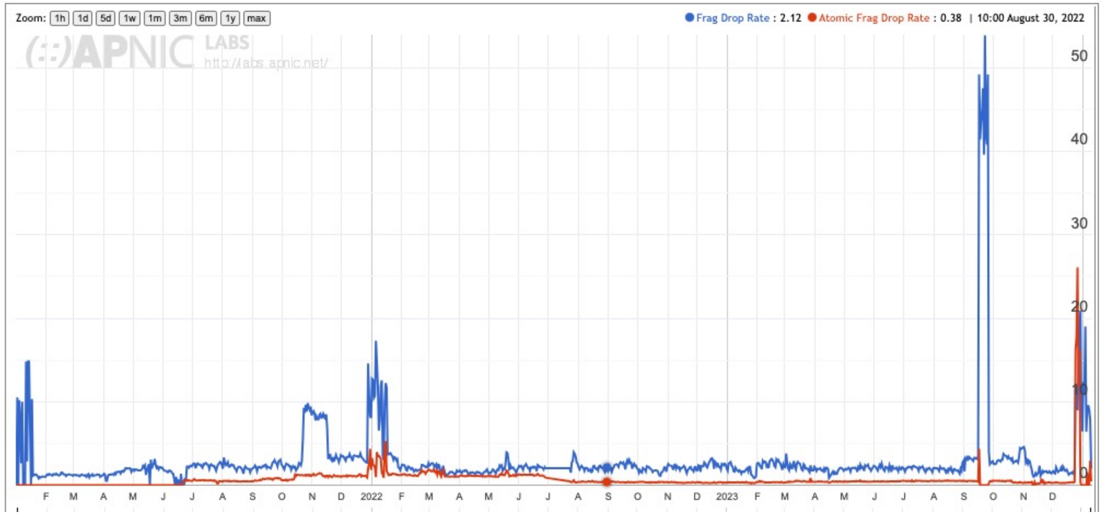
Use of V6FRAG Drop Rate for Australia (AU)



Use of V6FRAG Drop Rate for Japan (JP)



Use of V6FRAG Drop Rate for India (IN)



What to do?

- Set authoritative servers and recursive resolvers to override the EDNS Buffer Size in the query and respond with the TC bit set if the response is > 1232 bytes ?
- Configure stub clients to use DoH ?
- Revive the Additional Truncated Response draft? (https://datatracker.ietf.org/doc/html/draft-song-atr-large-resp-00)