IPv6 Operational Issues (with DNS)

Geoff Huston

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

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 [illegible] technology, and using IPv6 as the transport [illegible] efficient and as fast as using IPv4

Is this really true?

IPv6 and the DNS

How well is IPv6 supported in the DNS?

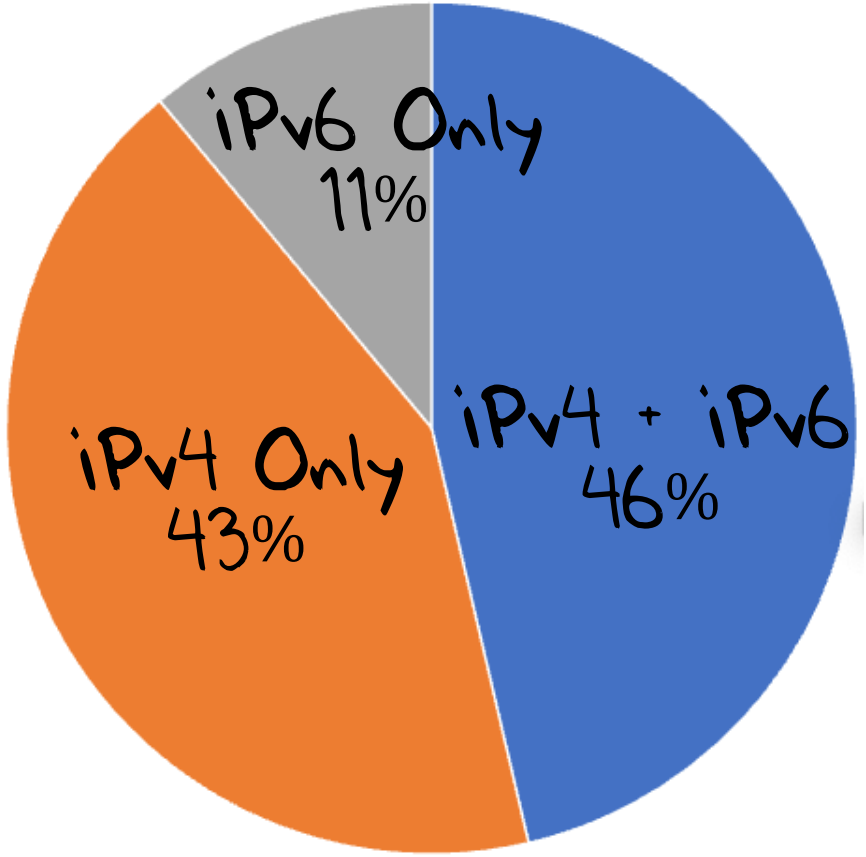
1. How does the DNS handle dual-stacked authoritative servers?
   - Is there a “happy eyeballs” version of DNS server selection?
   - Or is there a reverse bias to use IPv4?
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you?
3. And what about DNSSEC?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

Dual Stack and the 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 individual measurements



% of user measurements

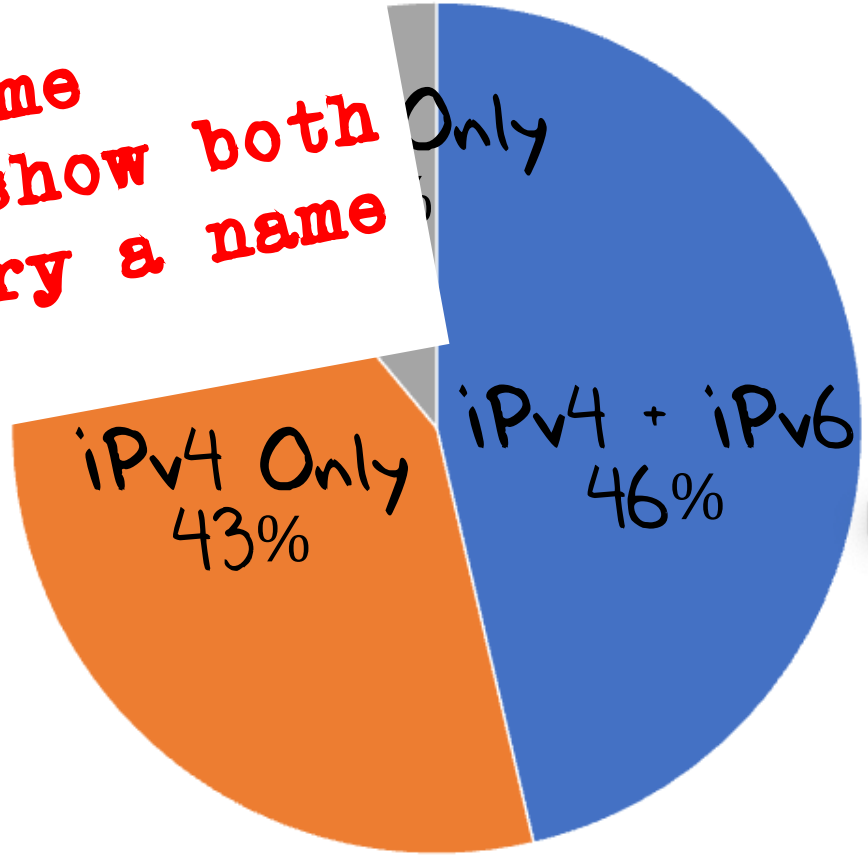
Dual Stack and the DNS

A “happy eyeballs*” DNS approach would be to prefer to use the IPv6 address

A “re[illegible]approach would be to prefer to use the IPv4 address

Less than one half of all name resolution query sequences show both protocols being used to query a name at the authoritative server

Data collected Dec 23 – Jan 24 using 445M individual measurements

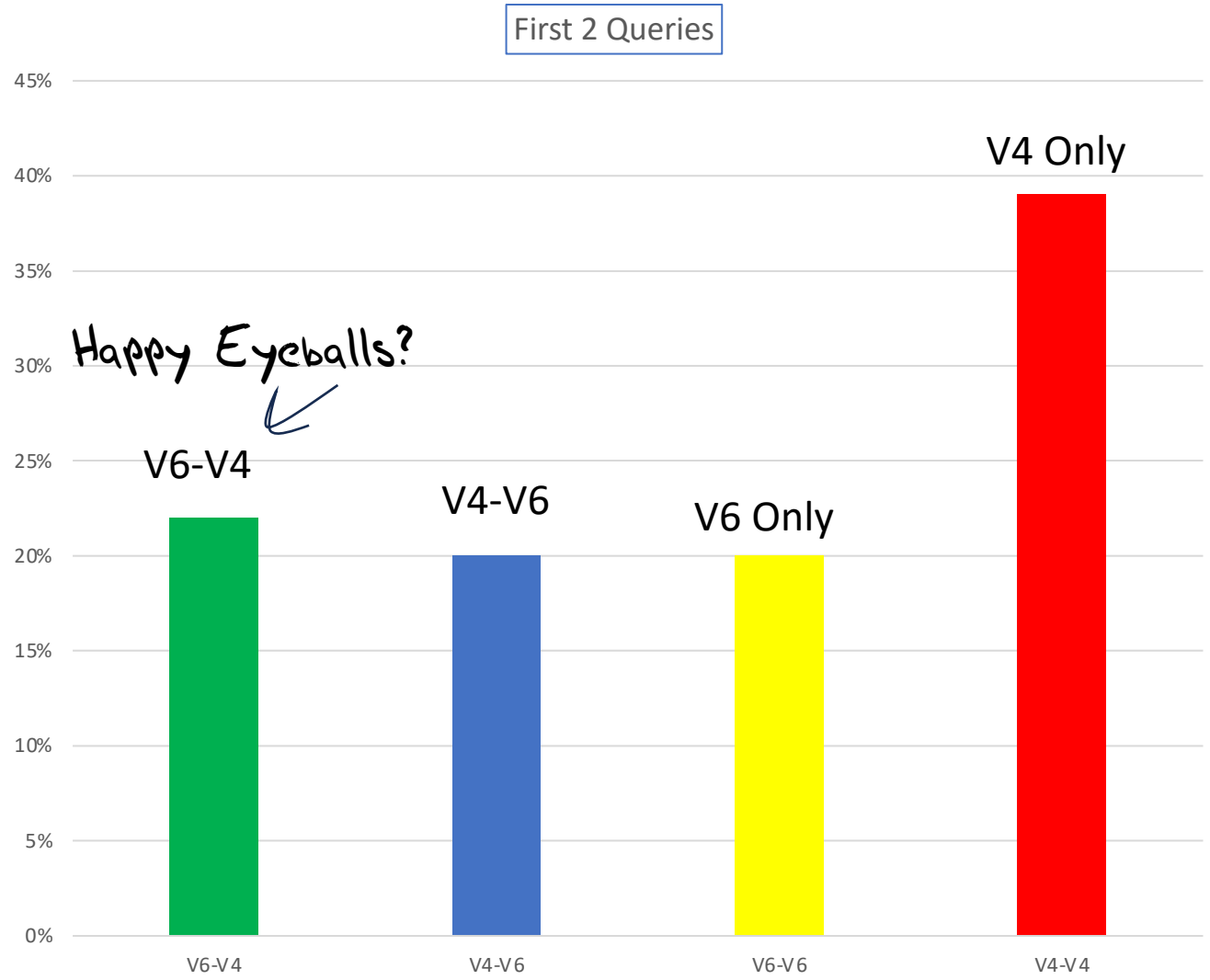


% of user measurements

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

We just don’t observe a visible bias to this “IPv6 First” approach

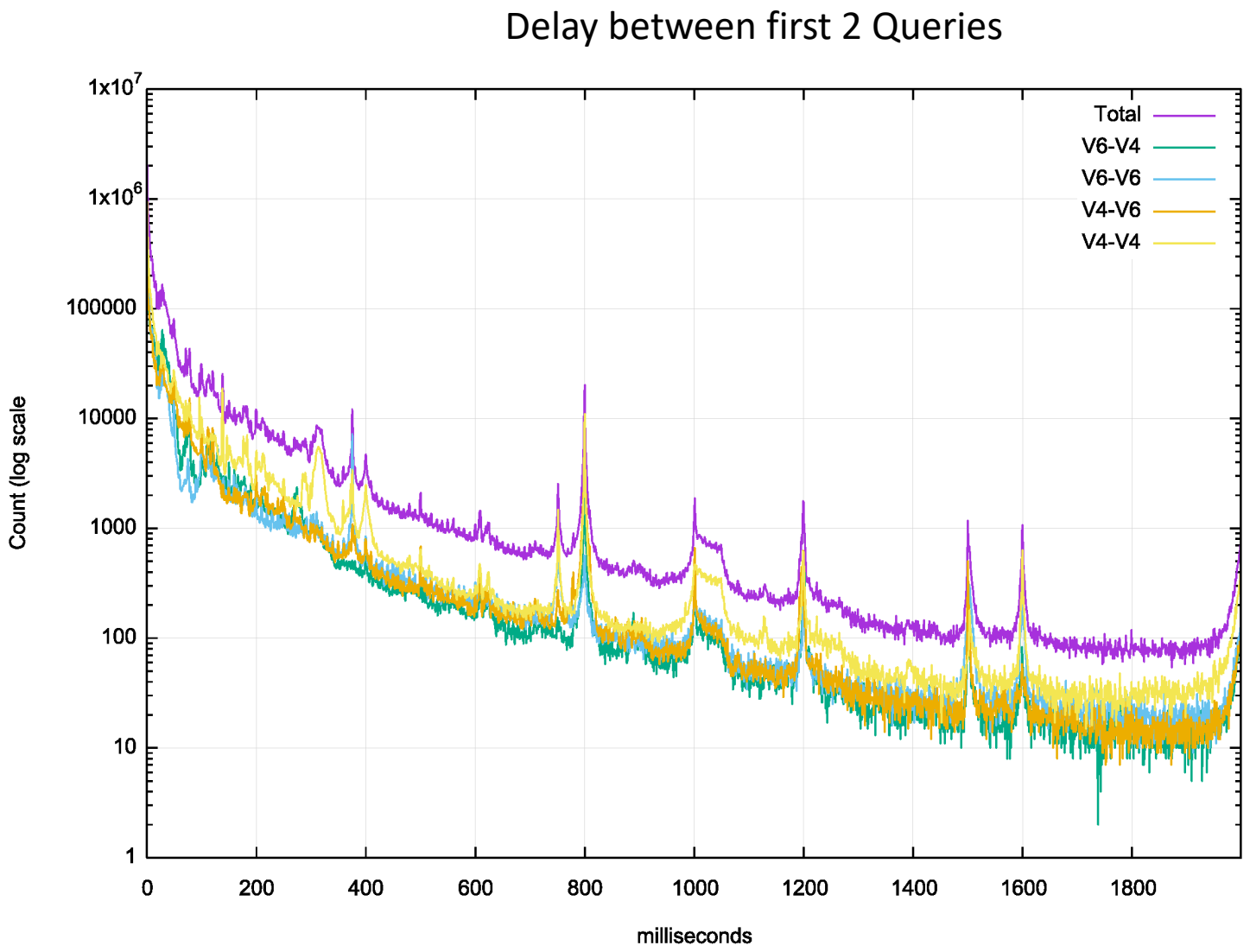


Dual Stack DNS

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

Which is observed in the data, but we also see evidence of conventional DNS timeout values of 370ms, 400ms, 800ms and 1 sec

Is the high repeat query count in the first 50 ms due to DNSMASQ behaviour?



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? Probably!
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

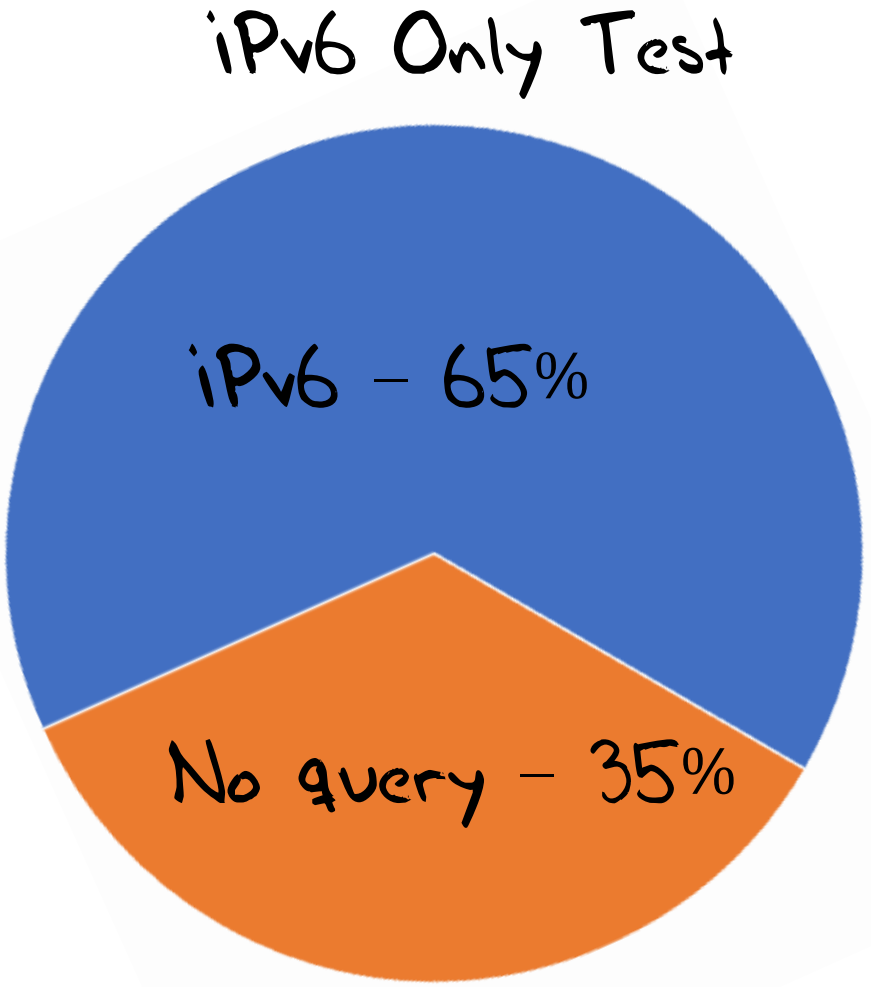
How well is IPv6 supported in the DNS?

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? Probably!
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?

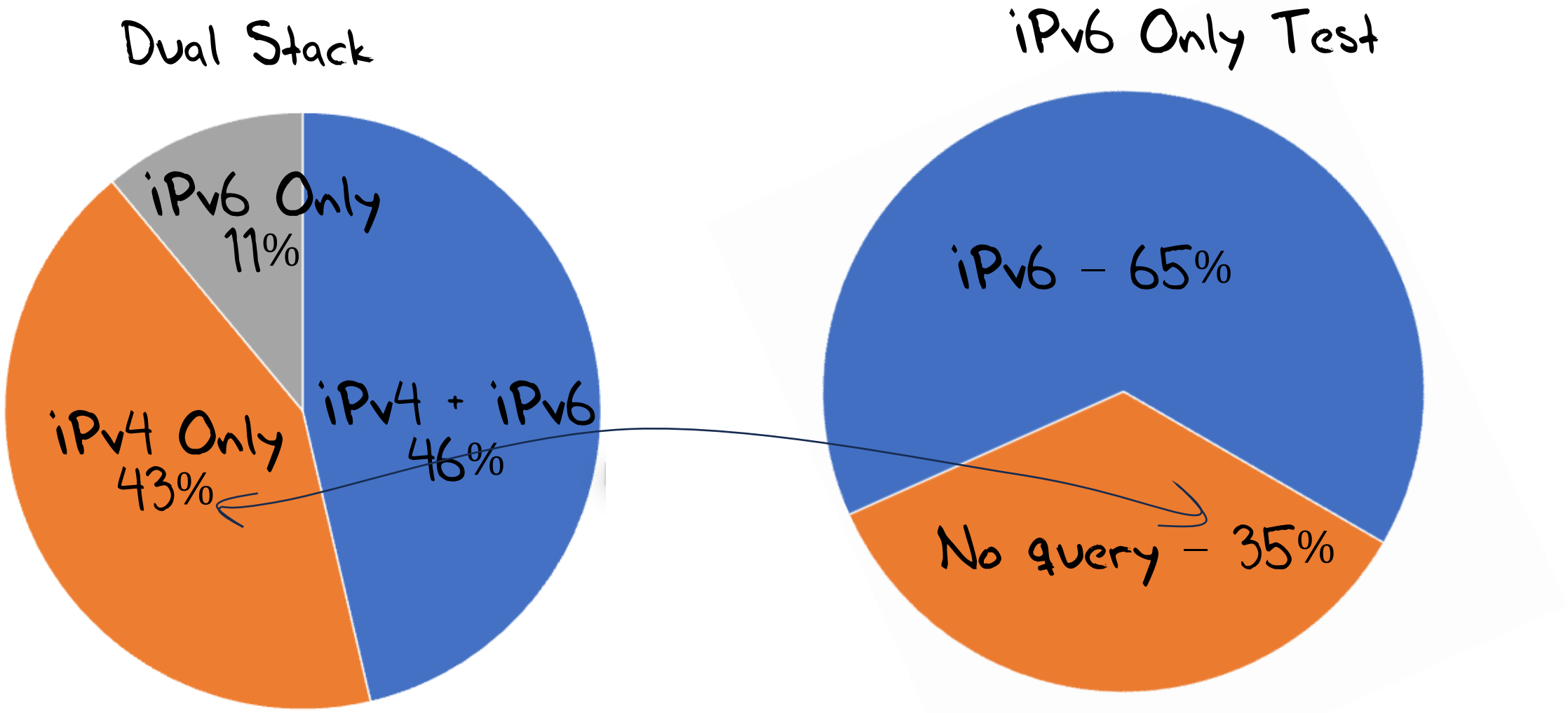
Dual Stack vs IPv6 only DNS

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? Probably!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you? Only 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? Probably!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you? Only 55%
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets?

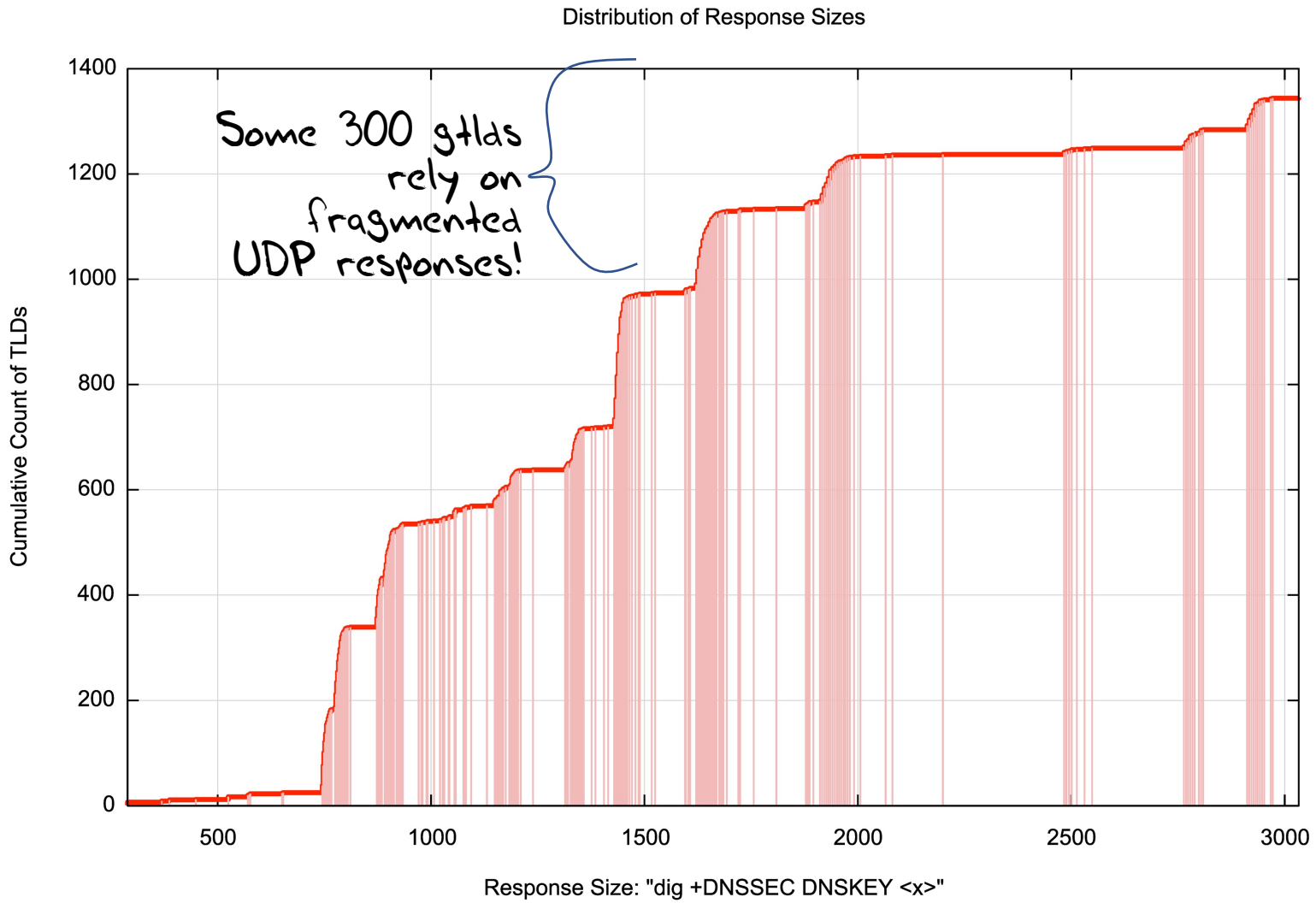
Who uses large DNS packets anyway?

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?



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?

  Yes!

  By sending large (>1500 octet) responses in the DNS and obeying the query's EDNS buffer size and fragmenting or truncating as determined by the query

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

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? Probably!
2. If you placed authoritative servers on an IPv6-only service how many users would be able to reach you? Only 55%
3. And what about DNSSEC?
   - How well does IPv6 support large UDP packets? Very Badly!

What should we do about this?

What can we do about it?

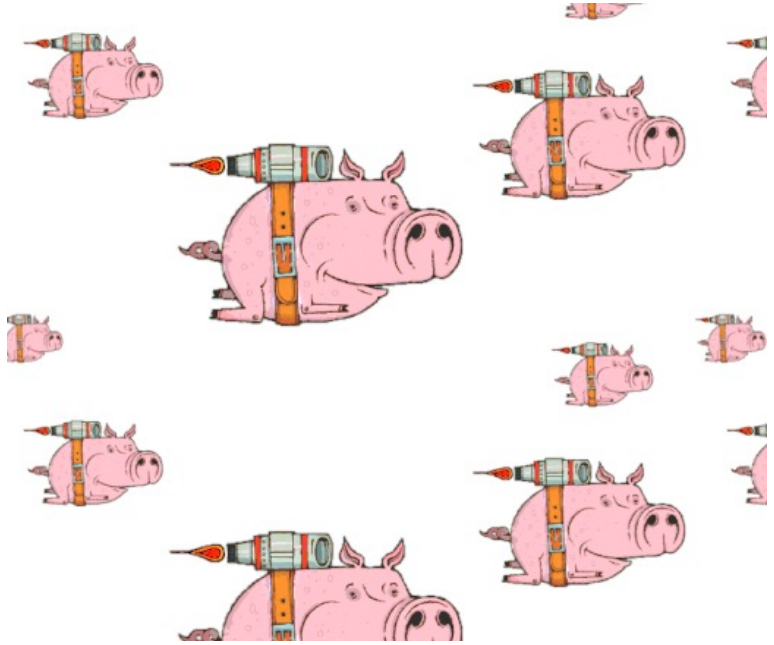
Fix it!

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 application behaviour to avoid the use of packet fragmentation completely



What do the RFC's say?

What do the RFC's say?

Internet Engineering Task Force (IETF)
Request for Comments: 8085
BCP: 145
Obsoletes: 5405
Category: Best Current Practice
ISSN: 2070-1721

L. Eggert
NetApp
G. Fairhurst
University of Aberdeen
G. Shepherd
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)
Request for Comments: 8085
BCP: 145
Obsoletes
Category:
ISSN: 207

L. Eggert
NetApp

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]

Abstract

The Use
transp
docume
applic
contro
provid
reliab
Congest

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?

DON'T FRAGMENT!

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

L. Eggert
NetApp

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

Applications that do not
discovery SHOULD
in IP
is
tha
in
firs

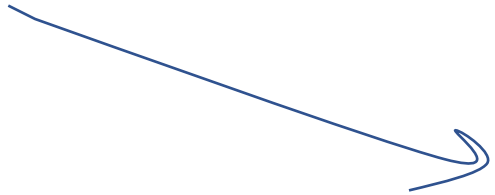
PMTUD
sult
MTU
s
(EMTU_S
576 bytes and the
EMTU_S is 1280 bytes [RFC2460].

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

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/

Is the DNS ready for IPv6-only?

Not yet!

Thanks!