# Measuring DNS Flag Day 2020

Geoff Huston
Joao Damas
APNIC Labs

# DNS Flag Day 2020

## DNS flag day 2020

## Contents



## What's next?

The next DNS Flag Day is scheduled for 2020-10-01. It focuses on the operational and security problems in DNS caused by Internet Protocol packet fragmentation.

# DNS Flag Day 2020

## DNS flag day 2020

## Contents



## What's next?

The next DNS Flag Day is scheduled for 2020-10-01. It focuses on the operational and security problems in DNS caused by Internet Protocol packet fragmentation.

## Action: Authoritative DNS Operators

If you are an authoritative DNS server operator, what you should do to help with these issues is ensure that your DNS servers can answer DNS queries over TCP (port 53). *Check your firewall(s) as well,* as some of them block TCP/53.

You should also configure your servers to negotiate an EDNS buffer size that will not cause fragmentation. The value recommended here is 1232 bytes.

*Authoritative DNS servers **MUST NOT** send answers larger than the requested EDNS buffer size!*

## Action: DNS Resolver Operators

Requrirements on the resolver side are more or less the same as for authoritative: ensure that your servers can answer DNS queries over TCP (port 53), and configure an EDNS buffer size of 1232 bytes to avoid fragmentation. Remember to check your firewall(s) for problems with DNS over TCP!

Most importantly: *Resolvers **MUST** resend queries over TCP if they receive a truncated UDP response (with TC=1 set)!*

# DNS Flag Day 2020

## DNS flag day 2020

## Contents



## What's next?

The next DNS Flag Day is scheduled for 2020-10-01. It focuses on the operational and security problems in DNS caused by Internet Protocol packet fragmentation.

## The exact date

**2020-10-01** (October 1st 2020)

## Action: Authoritative DNS Operators

If you are an authoritative DNS server operator, what you should do to help with these issues is ensure that your DNS servers can answer DNS queries over TCP (port 53). *Check your firewall(s) as well,* as some of them block TCP/53.

gotiate an EDNS buffer size that will not
ed here is 1232 bytes.

*answers larger than the requested EDNS*

## Action: DNS Resolver Operators

Requrirements on the resolver side are more or less the same as for authoritative: ensure that your servers can answer DNS queries over TCP (port 53), and configure an EDNS buffer size of 1232 bytes to avoid fragmentation. Remember to check your firewall(s) for problems with DNS over TCP!

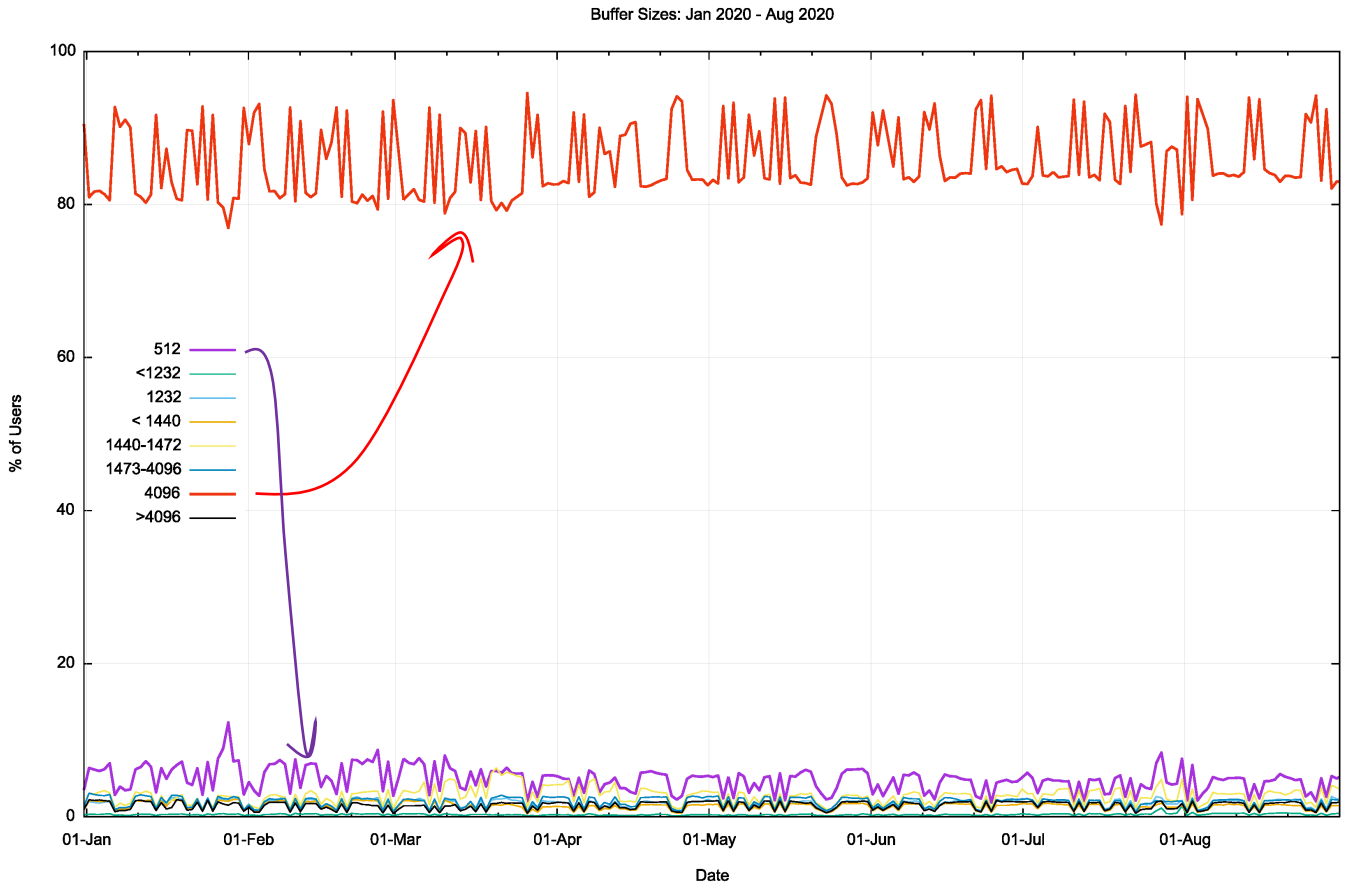
Most importantly: *Resolvers **MUST** resend queries over TCP if they receive a truncated UDP response (with TC=1 set)!*

# What Happened?

We'd like to look at two aspects of this work:

- What happened on 1 October 2020 (and thereafter) in the DNS?
- Is that recommended value of 1,232 just right? Too small? Too large?

# Looking at EDNS(0) Buffer Sizes

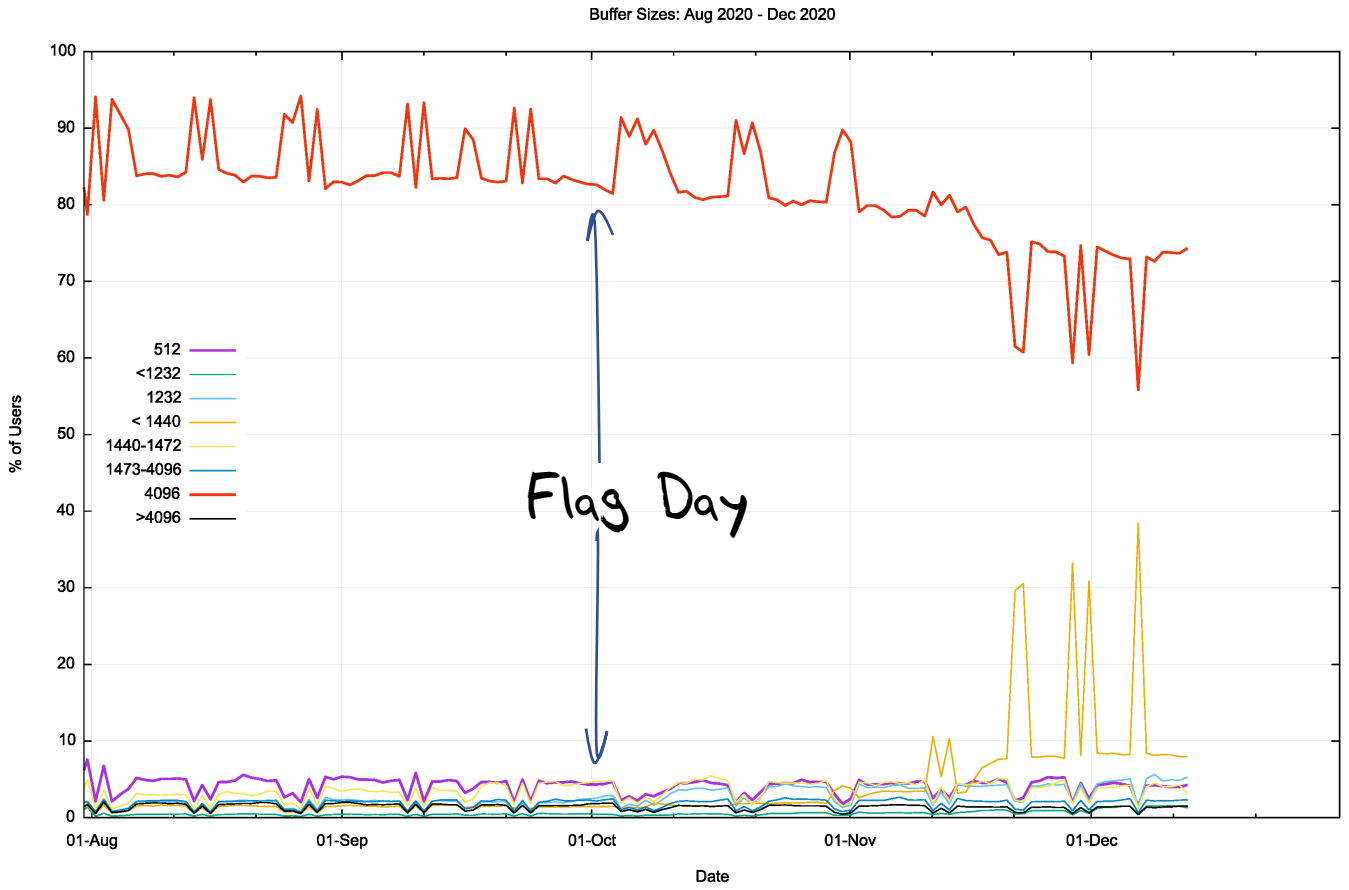



Jan 2020 – August 2020

- 4,096 used by queries from 80% - 95% of users
- 512 (no size specified) used by 10% of users
- Weekday / Weekend profile suggesting a difference between enterprise and access ISP profiles

These results are from looking at queries between recursive resolvers and authoritative servers

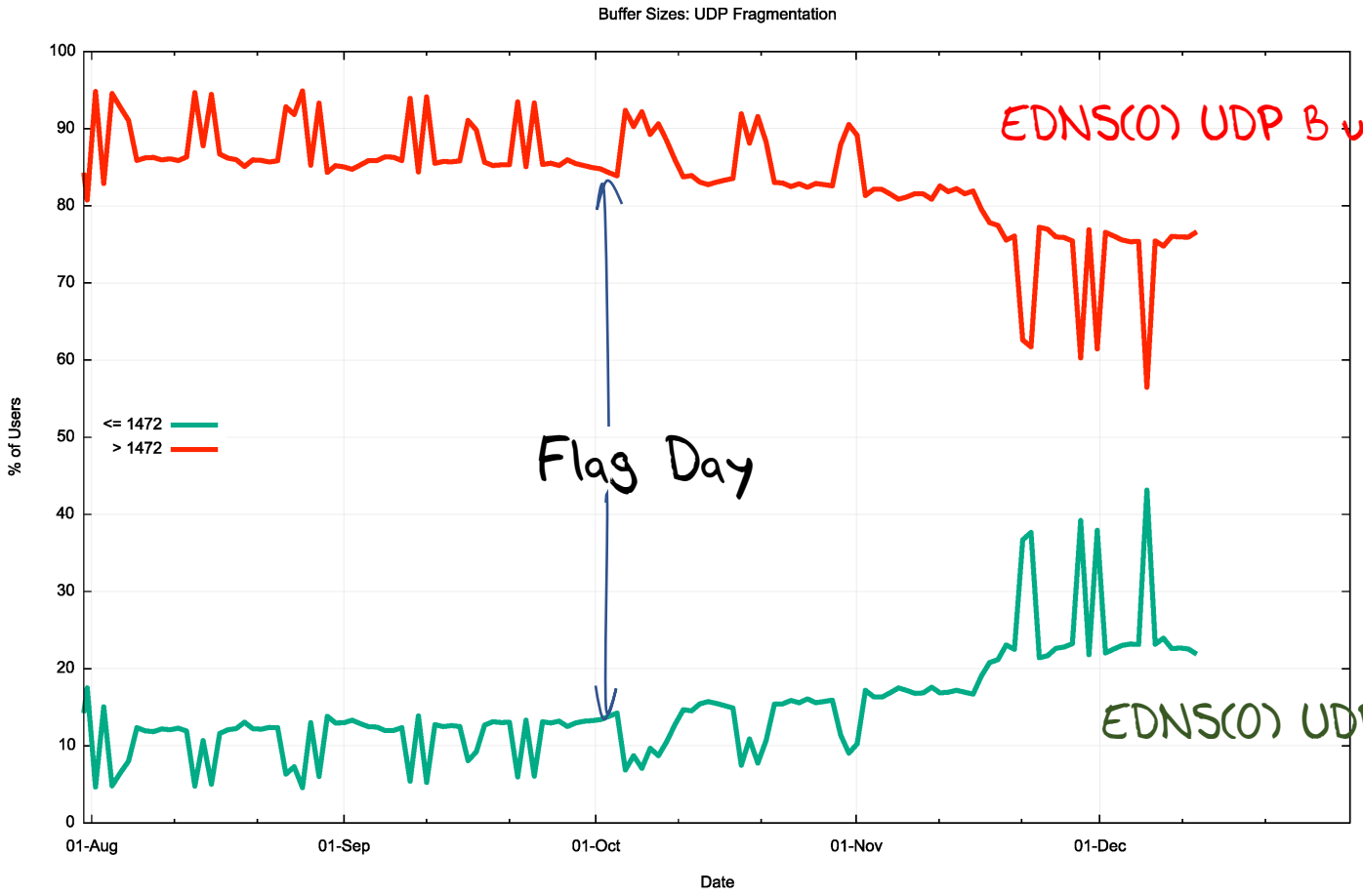
# Flag Day 2020




August 2020 – December 2020

- Use of 4,096 buffer size dropped from ~84% to 70% of users by December
- Rise in 1,400 buffer size  to 8% of users

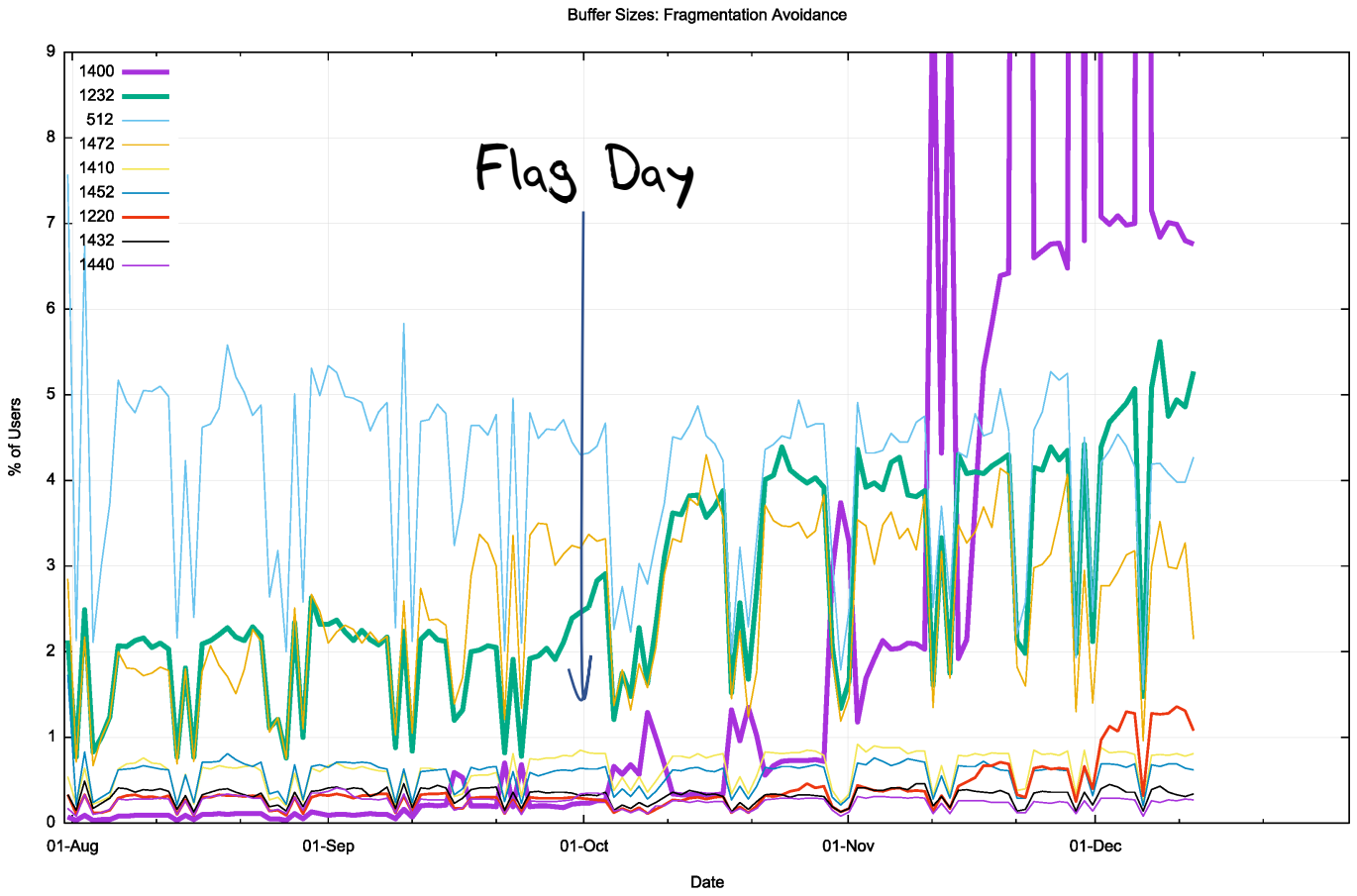
# UDP Fragmentation




August 2020 – December 2020

- Fragmentation avoidance settings rose from 12% of users to ~22% of users

# UDP Fragmentation Avoidance




August 2020 – December 2020

- 1,232 is now used by 5% of users
- 1,400 is now used by 7% of users
- 284 different sizes between 512 and 1472 observed in this data set

# Pick a Size

- Is there a “right” size for this parameter?
- What are we attempting to achieve here when trying to select the threshold point to get the DNS to switch to use TCP?
- Should we use a low value and switch “early”?
- Should we use a high value and switch “late”?

# IP and Packet Sizes

| | IPv4 | IPv6 |
|---|---|---|
| Minimum IP Packet Size | 20 | 40 |
| Maximum Assured Unfragmented Packet Size | 68 | 1,280 |
| Assured Host Packet Size | <= 576 | <= 1,500 |
| Maximum Packet Size | 65,535 | 65,575* |

*4,294,967,336 (Jumbogram)

# Some Questions

- Why choose 1,232 octets as the threshold point to truncate a UDP response in Flag Day 2020?
- How bad is UDP Fragmentation loss in the DNS?
- How bad is TCP in the DNS?

# Mesurement Challenges

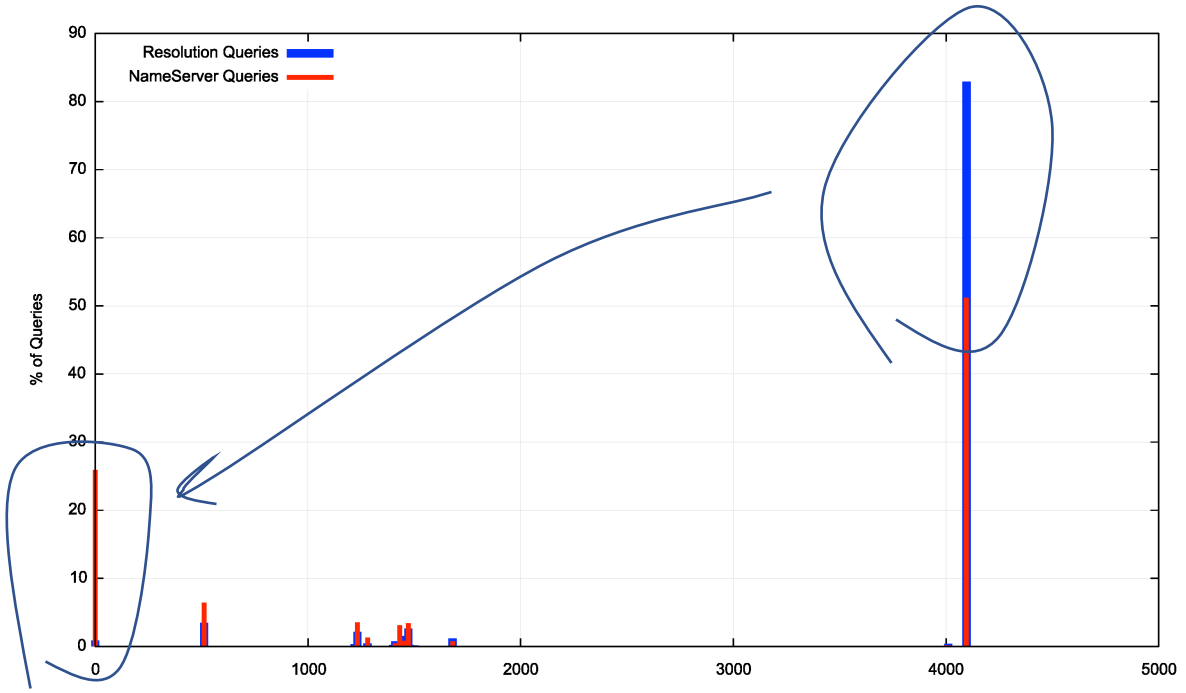
- How to perform a large scale measurement?
  - We embed the measurement in an advertisement to distribute the measurement script to a broad set of test cases
- How to detect DNS resolution success?
  - We use a technique of “glueless” delegation to force a resolve to explicitly resolve the name of a  name server – a successful resolution is signalled by the resumption of the original resolution task
- How to characterise DNS behaviour?
  - We pad the response to create the desired response size. Each test uses a response size selected at random from 11 pad sizes. We also use an unpadded short response as a control

# Limitations

- We are measuring the DNS path between recursive resolvers and the authoritative name servers. This is a measurement of the “interior” of the Internet. It is not a measurement of the stub-to-recursive paths at the edge of the network.
- Some resolvers alter their behaviour when resolving name server names
  - In some 30% of cases the EDNS(0) Buffer Size is either dropped from the query, or dropped below 1452 octets

# Limitations

- In some 30% of cases the EDNS(0) Buffer Size is either dropped from the query, or dropped below 1452 octets



# "Base Test" September 2020

| Size | Tests | | Passed | Failed | Rate |
|---|---|---|---|---|---|
| 1230 | 4,303,845 | | 4,282,457 | 21,388 | 0.50% |
| 1270 | 4,308,667 | | 4,287,046 | 21,621 | 0.50% |
| 1310 | 4,307,456 | | 4,286,064 | 21,392 | 0.50% |
| 1350 | 4,304,230 | | 4,282,752 | 21,478 | 0.50% |
| 1390 | 4,310,182 | | 4,288,413 | 21,769 | 0.51% |
| 1430 | 4,303,906 | | 4,281,858 | 22,048 | 0.51% |
| 1470 | 4,308,722 | | 4,269,785 | 38,937 | 0.90% |
| 1510 | 4,303,923 | | 4,197,910 | 106,013 | 2.46% |
| 1550 | 4,306,824 | | 4,194,465 | 112,359 | 2.61% |
| 1590 | 4,300,559 | | 4,187,575 | 112,984 | 2.63% |
| 1630 | 4,305,525 | | 4,191,994 | 113,531 | 2.64% |

Onset of server UDP fragmentation

# TCP behaviour

This selects the subset of cases where the recursive resolver was passed a truncated UDP response, which should trigger the resolver to use TCP

Truncated UDP response, no followup TCP

Stalled TCP session with missing ACK from data segment

Completed TCP session but no signal of resumption of original resolution

| Size | TCP Use | | Pass | Fail | | NO TCP | NO ACK | TCP OK |
|---|---|---|---|---|---|---|---|---|
| 1230 | 9% | | 98.7% | 1.3% | | 11.4% | 28.3% | 60.3% |
| 1270 | 13% | | 99.0% | 1.0% | | 12.2% | 27.7% | 60.1% |
| 1310 | 13% | | 99.0% | 1.0% | | 13.2% | 26.4% | 60.4% |
| 1350 | 13% | | 99.0% | 1.0% | | 13.0% | 27.9% | 59.0% |
| 1390 | 14% | | 99.0% | 1.0% | | 15.2% | 27.1% | 57.7% |
| 1430 | 14% | | 99.1% | 0.9% | | 15.7% | 25.9% | 58.5% |
| 1470 | 30% | | 98.5% | 1.5% | | 9.2% | 58.3% | 32.5% |
| 1510 | 36% | | 98.1% | 1.9% | | 22.7% | 47.2% | 30.1% |
| 1550 | 36% | | 98.1% | 1.9% | | 23.2% | 46.8% | 30.0% |
| 1590 | 36% | | 98.1% | 1.9% | | 24.5% | 45.7% | 29.8% |
| 1630 | 36% | | 98.1% | 1.9% | | 25.6% | 45.5% | 28.9% |

Responses which are larger than 1,430 octets show a higher loss rate

# TCP behaviour

TCP shows a base failure rate of some 1% to 2% of tests

- For smaller responses this may be due to enthusiastic filtering of TCP port 53 packets
- For larger responses TCP “Black Hole” factors may be involved, as the server was configured to use a local 1,500 octet MTU and maximum size TCP data segments may have triggered Path MTU pathologies

# Forcing TCP

- Here we set the server's max buffer size to 512, forcing all resolution attempts to use TCP

| DNS Response Size | Tests | TCP Pass Rate | TCP Fail Rate | IPv4 Failure Rate | IPv6 Failure Rate |
|---|---|---|---|---|---|
| 1150 | 1,104,539 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1190 | 1,105,126 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1230 | 1,105,601 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1270 | 1,104,571 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1310 | 1,104,521 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1350 | 1,104,068 | 98.5% | 1.6% | 2.0% | 1.6% |
| 1390 | 1,105,080 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1430 | 1,104,527 | 98.5% | 1.6% | 1.9% | 1.6% |
| 1470 | 1,103,423 | 98.3% | 1.8% | 2.1% | 1.8% |
| 1510 | 1,104,960 | 98.3% | 1.8% | 2.1% | 1.8% |
| 1550 | 1,105,566 | 98.3% | 1.8% | 2.1% | 1.8% |
| 1590 | 1,103,609 | 98.3% | 1.8% | 2.1% | 1.8% |
| 1630 | 1,106,284 | 98.3% | 1.8% | 2.1% | 1.8% |

iPv4 shows a slightly higher failure rate than iPv6

# UDP behaviour

This selects the subset of cases where the recursive resolver was not passed a truncated UDP response and did not attempt a TCP connection

| Size | UDP Use | Pass | Fail |
|---|---|---|---|
| 1230 | 91% | 99.6% | 0.4% |
| 1270 | 87% | 99.6% | 0.4% |
| 1310 | 87% | 99.6% | 0.4% |
| 1350 | 87% | 99.6% | 0.4% |
| 1390 | 86% | 99.6% | 0.4% |
| 1430 | 86% | 99.6% | 0.4% |
| 1470 | 70% | 99.4% | 0.6% |
| 1510 | 64% | 97.2% | 2.8% |
| 1550 | 64% | 97.0% | 3.0% |
| 1590 | 64% | 97.0% | 3.0% |
| 1630 | 64% | 97.0% | 3.0% |

Onset of server UDP fragmentation

# UDP behaviour

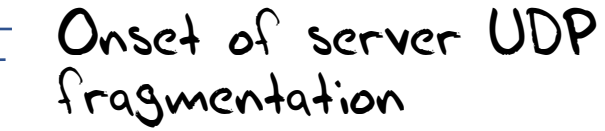
UDP shows a base failure rate of some 0.5% to 3% of tests

- For smaller responses this may be due to residual filtering of UDP port 53 packets greater than 512 octets in size
- For larger responses UDP fragmentation is the likely factor where the buffer size permits the server to transmit fragmented UDP packets, but they appear not to reach the resolver client

# Forcing UDP

- Here we alter the server to treat all queries as if they had signalled a buffer size of 4,096 octets

| DNS Response Size | Tests | UDP Pass Rate | UDP Fail Rate | IPv4 Failure Rate | IPv6 Failure Rate |
|---|---|---|---|---|---|
| 1150 | 1,140,192 | 99.6% | 0.4% | 0.6% | 0.1% |
| 1190 | 1,138,792 | 99.6% | 0.4% | 0.6% | 0.1% |
| 1230 | 1,273,730 | 99.6% | 0.4% | 0.6% | 0.1% |
| 1270 | 1,272,765 | 98.1% | 1.9% | 2.4% | 1.2% |
| 1310 | 1,275,436 | 98.2% | 1.8% | 2.4% | 1.2% |
| 1350 | 1,272,634 | 98.2% | 1.8% | 2.4% | 1.2% |
| 1390 | 1,273,332 | 98.1% | 1.9% | 2.4% | 1.2% |
| 1430 | 1,274,189 | 97.8% | 2.2% | 2.6% | 1.6% |
| 1470 | 1,274,581 | 96.9% | 3.1% | 3.7% | 17.6% |
| 1510 | 1,273,496 | 85.0% | 15.0% | 14.2% | 17.6% |
| 1550 | 1,274,776 | 85.0% | 15.0% | 14.4% | 17.7% |
| 1590 | 1,276,441 | 85.1% | 14.9% | 14.4% | 17.6% |
| 1630 | 1,275,233 | 85.1% | 14.9% | 14.5% | 17.6% |



# Forcing UDP

- A number of resolvers will discard a DNS response if it is larger than the original buffer size
  - This appears to occur in some 2% - 3% of cases
- A number of resolvers do not receive fragmented UDP packets
  - This appears to occur in ~11% of cases in IPv4, and ~15% of cases in IPv6

# DNS Flag Day 2020

We appear to have repurposed the EDNS(0) Buffer Size parameter

- It was originally designed as a signal from the client to the server of the client's capability to receive a DNS response over UDP
  - Oddly enough no comparable signal was defined for TCP, even though, presumably, the same client-side memory limitations for DNS payloads would exist
- It appears to have been intended as a UDP mechanism that "can help improve the scalability of the DNS by avoiding widespread use of TCP for DNS transport." (RFC 6891)
- The Flag Day measures appear to repurpose this parameter as a **UDP fragmentation avoidance signal**

# DNS Transport Considerations

- Unfragmented UDP is relatively fast, stable and efficient
  - There is a slight increase in drop rates above 512 octets to around 0.5%
  - There is no visible change in drop rates in payloads up to 1500 octets in size
- Fragmented UDP has a very high drop rate
  - Between 11% and 15% drop rate in IPv4 and IPv6 respectively
  - It is more likely to be due to security filtering practice, although no specific fragmentation measurement has been made
- TCP is less efficient and slower than unfragmented UDP, but far better in performance terms than Fragmented UDP
  - Base failure rate for TCP is between 1% to 2% of cases

# DNS Transport Priorities

- Use unfragmented UDP as much as possible
- Avoid dynamic discovery of path MTU / fragmentation onset
- Prefer TCP over responding with fragmented UDP for larger responses

# Buffer Size Considerations

- One size fits all?
  - 1232 is a conservative value with a high assurance of fragmentation avoidance
  - Early onset of TCP extracts a marginal cost in terms of efficiency and speed of resolution
  - Could we improve on this by tailoring the value to suit the context of the query/response transaction?
- Customised settings
  - Fragmentation onset occurs in different ways on different paths
  - Our measurements suggest that in the “interior” of the Internet between recursive resolvers and authoritative servers the prevailing MTU is at 1,500. There is no measurable signal of use of smaller MTUs in this part of the Internet *
  - Fragmentation onset occurs differently for IPv4 and IPv6



* The “edge” of the internet is likely to be different – no measurements were made for edge scenarios in this study

# For Recursive to Authoritative

Our measurements suggest setting the EDNS(0) Buffer size to:

| | |
|---|---|
| **IPv4** | **1,472 octets** |
| **IPv6** | **1,452 octets** |

A small additional performance improvement can be made by using a lower TCP MSS setting – our measurements of a 1,200 octets setting showed a small but visible improvement in TCP resilience for large (multi-segment) payloads. In the TCP the marginal cost of a highly conservative setting for the MSS is far lower than the cost of correcting MTU issues.

# Thanks!

Full Report: https://www.potaroo.net/ispcol/2020-11/xldns.html (part 1)
https://www.potaroo.net/ispcol/2020-12/xldns2.html (part 2)