Cause and Effect

Geoff Huston

APNIC Labs

July 2020

Approaches to DNS Measurement...

Gather a set of DNS queries and/ or responses from a recursive resolver or an authoritative server

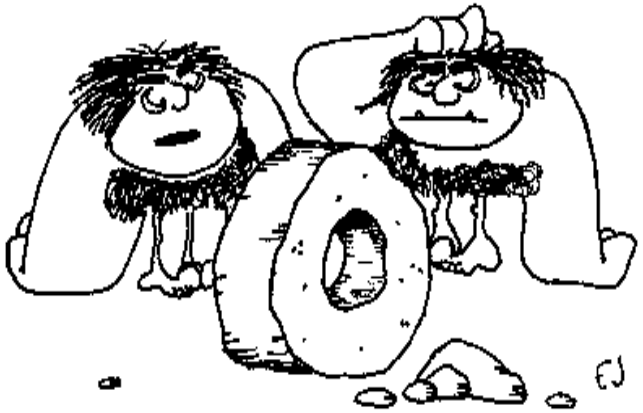
Sift through the query/response log using some form of selection criteria

Propose a theory of DNS behaviour based on the observed traffic

Approaches to DNS Measurement...

The problem with this approach is that there is no clear picture of **why** the query was made:

- It could've been a stub resolver query from an application
- Or a resolver performing cache refresh
- Or a query log analyzer performing query/response validation
- Or some form of attack or backscatter from an attack
- Or <other>!



Alternatively…

We might understand the *effect* better if we controlled the *cause*

i.e. generate queries in a known context and look at their effect within in the DNS resolution environment

Inside looking Out

Instrument a DNS client

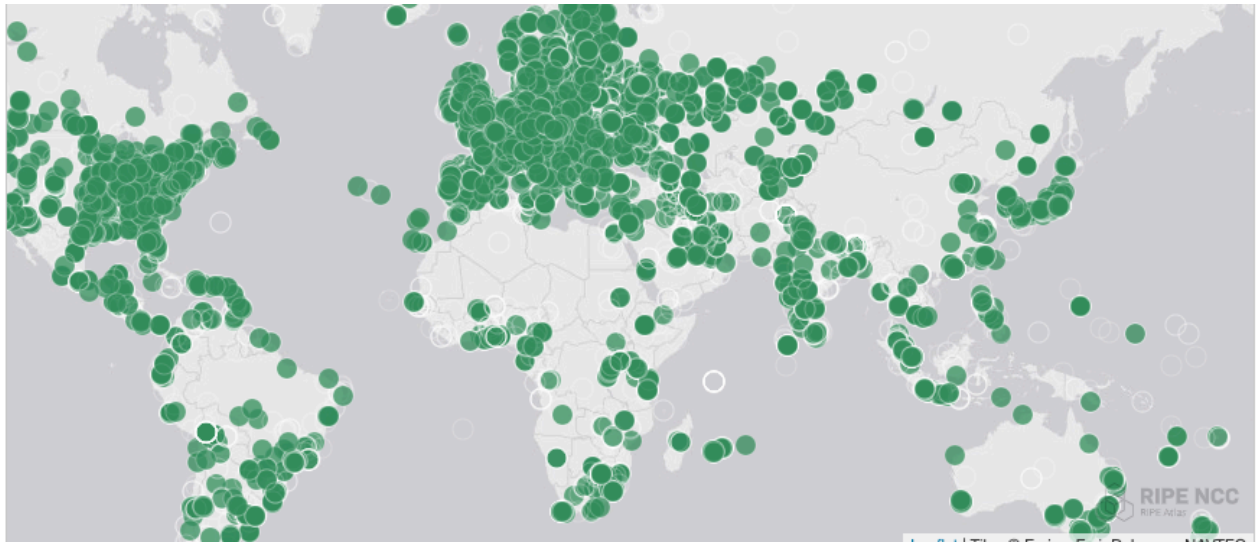
- Use the client to generate various DNS queries
- Measure the absolute outcomes and the variance

This needs the ability to either coopt or manufacture a collection of willing clients

Inside looking Out

RIPE Atlas

- Many thousands of end points installed in end user networks
- Programmable DNS queries
- Report back



Outside looking In

Set up authoritative server(s)

- Enroll end users to send queries to it
- Measure the outcomes from the perspective of the server, not the end client

How to measure using millions of end devices?

APNIC Lab's approach

- we originally wanted to measure IPv6 deployment as seen by end users
- We wanted to say something about ALL users
- So we were looking at a way to sample end users in a random but statistically significant fashion across the entire network
- We stumbled across the advertising networks...

APNIC Thank you for helping us measure the Internet.

Thank you for helping us measure the Internet.

```
<!DOCTYPE html>
<html><head data-gwd-animation-mode="quickMode"><meta name="GCD"
content="YTk3ODQ3ZWZhN2I4NzZmMzBkNTEwYjJl657daa7a9fa4c339ce298ace1f626e3e"/>

  <meta name="generator" content="Google Web Designer 1.2.1.0121">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <script type="text/javascript" src="https://s0.2mdn.net/ads/studio/Enabler.js"></script>
  </script>
  <script type="text/javascript" src="html5ad.js">
 </script>
<body>
  <img src="apnic-logo.png" style="float:left"/>
  Thank you for helping us measure the Internet.

  <script type="text/javascript">
    runLabsTests();
  </script>
  <!-- This section contains metadata about the ad. Most importantly, the ad size. -->
  <script type="text/gwd-admetadata">
{"version":1,"type":"DoubleClick","format":"","template":"","politeload":true,"counters":[],"timers":[],"exits":
[],"creativeProperties":{"minWidth":468,"minHeight":60,"maxWidth":468,"maxHeight":60},"components":[]}</script>
</body></html>
```

What can be scripted

Not much:

- http.FetchImg()

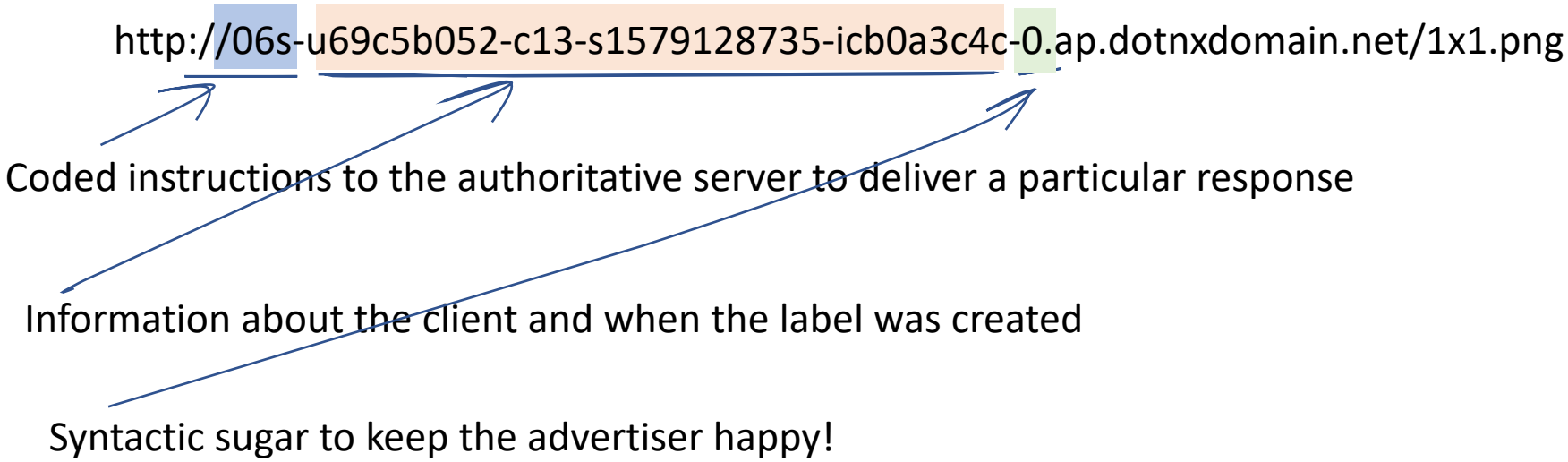
  i.e. attempt to retrieve a URL

But that's enough!

- It's EXACTLY what users do!
- A URL consists of a DNS question and an HTML question
- What if we point both the DNS and the HTML to servers we run?
- As long as each Ad execution uses unique names we can push the user query back to our servers

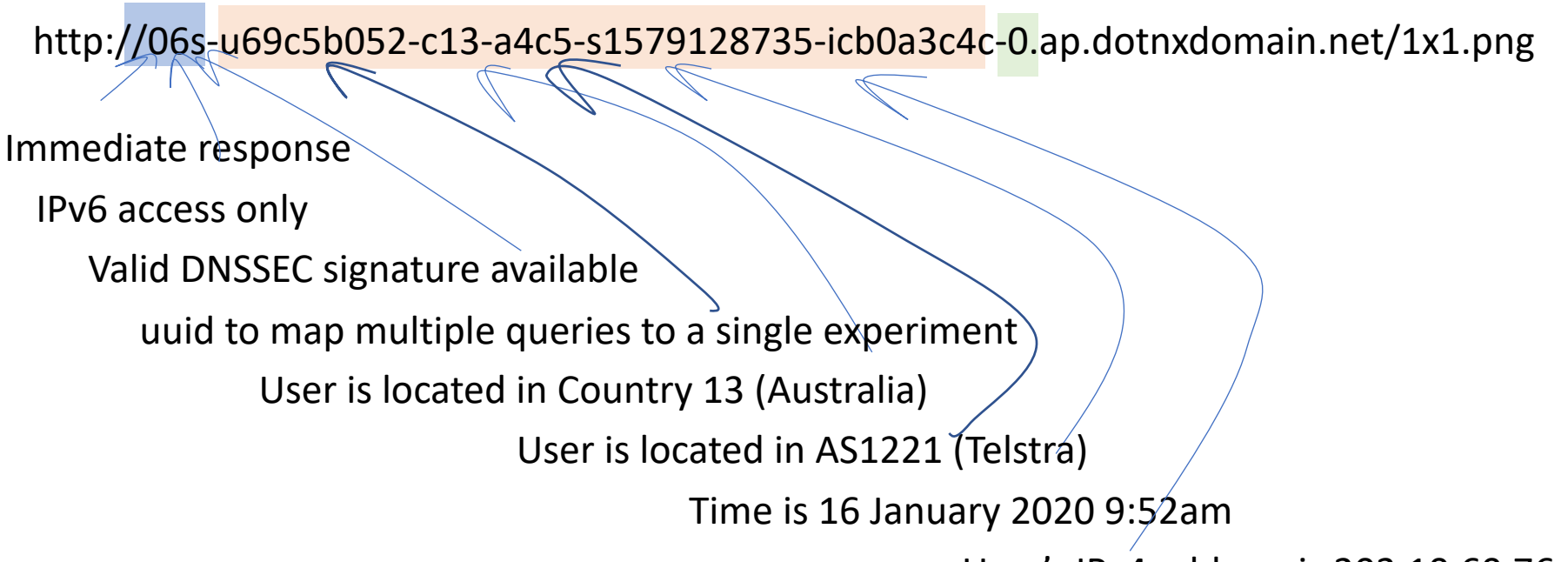
DNS Label encoding

Think of a DNS name as a microcoded instruction set directed to programmable DNS and HTTP servers …



DNS Label Encoding

Think of a URL name as a microcoded instruction set directed to programmable DNS and HTTP servers …



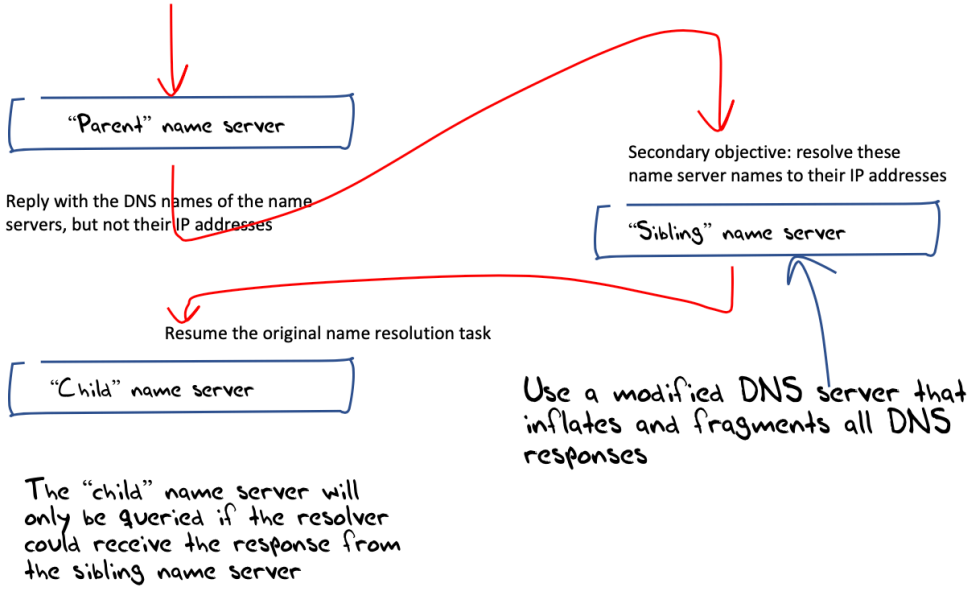
Generating a DNS response behaviour

Use the query label to trigger a particular DNS response behaviour, such as:

- Validly DNSSEC-signed or invalidly signed
- Padded (large) response
- Truncated response
- Delayed response
- SERVFAIL response
- NXDOMAIN

Confirming responses

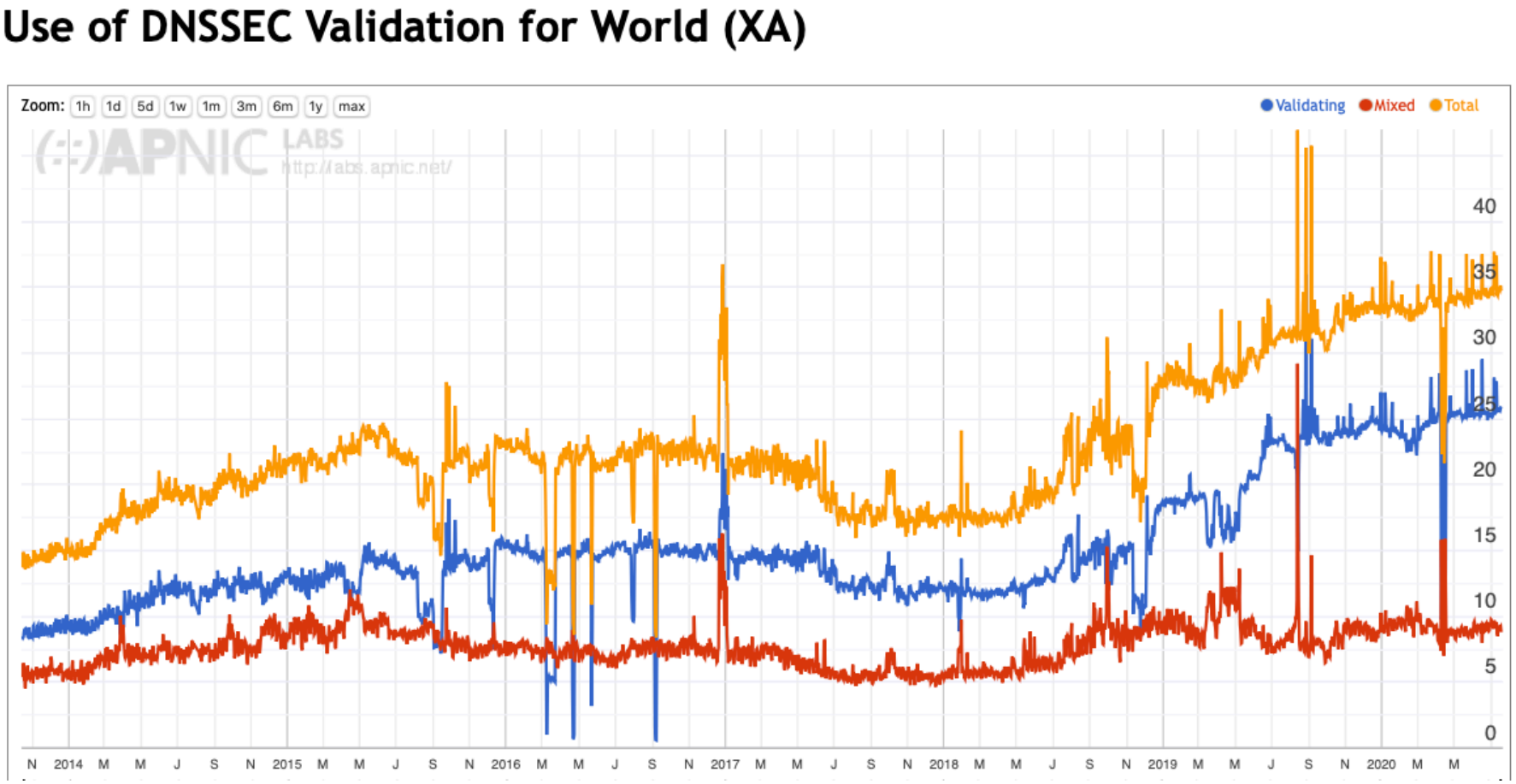
- How can you tell if the client received the DNS response?
  - Leverage the DNS discovery process using a technique of “glueless” delegation where the label is at the apex of its own delegated zone
  - Inject the DNS behaviour into the name server response
  - The final query will only be made if the discovery name server response was received



Example Measurements

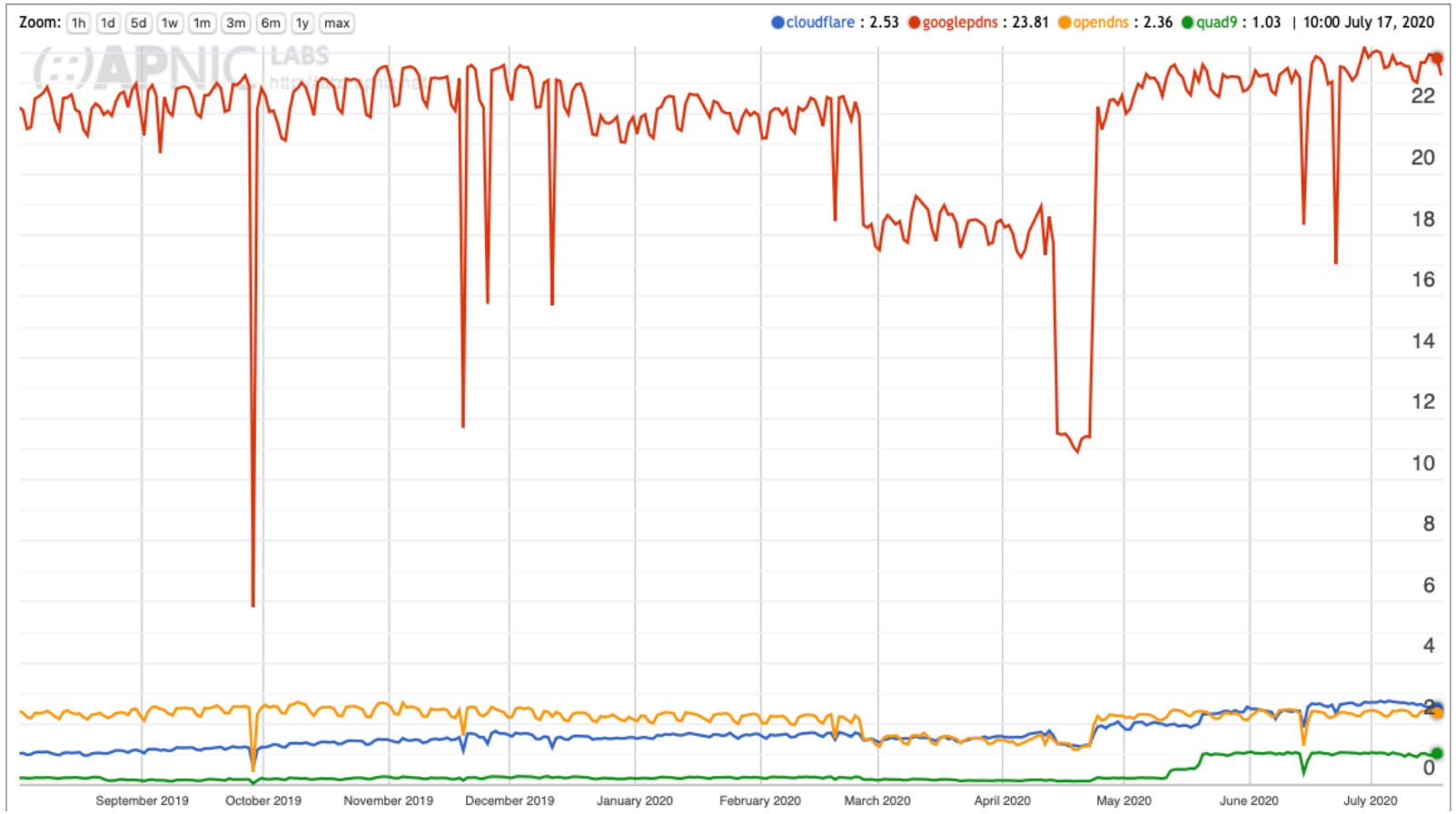
- DNSSEC validation capability
- Distribution of use of open DNS resolvers
- DNS happy eyeballs
- DNS over IPv6
- NSEC caching
- Qname minimisation
- Zombie query patterns
- KSK roll analysis

Example: DNSSEC Validation



https://stats.labs.apnic.net/dnssec/XA

Example: Open DNS Resolver Use



https://stats.labs.apnic.net/rvrs/XA

Thanks!