APNIC 38

What If Everyone Did It?

Geoff Huston

APNIC Labs

BRISBANE, AUSTRALIA

9-19 September 2014

#apnic38

DNSSEC and DNS Security

```
$ dig xxx.00001.z.dotnxdomain.net
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; ANSWER SECTION:
xxx.00001.z.dotnxdomain.net. 1          IN        A          199.102.79.186
```

What does this mean?

```
$ dig xxx.00001.z.dashnxdomain.net
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
;; ANSWER SECTION:
xxx.00001.z.dashnxdomain.net. 3600 INA          199.102.79.188
```

DNSSEC and DNS Security

```
$ dig xxx.00002.z.dotnxdomain.net
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 9216
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;xxx.00002.z.dotnxdomain.net.         IN          A

;; Query time: 619 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Sep 10 01:20:02 UTC 2014
;; MSG SIZE  rcvd: 56
```

What does this DNS response mean?

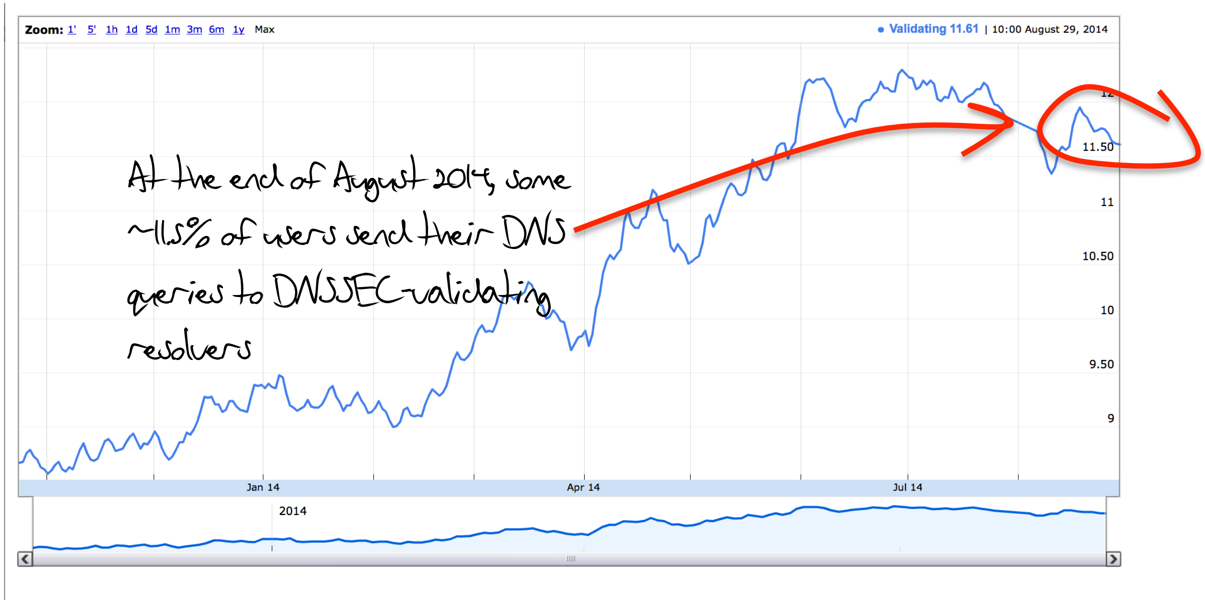
DNSSEC and DNS Security

- Setting the AD bit in a recursive resolver response seems like a rather unimpressive way of conveying a positive security outcome, and in the same manner, setting SERVFAIL seems like a rather poor way of conveying a failed security outcome
- Various approaches to securing the channel between the client and the recursive resolver have been suggested, but in a simple lightweight UDP transaction model this can be challenging
- Perhaps it would be simpler for the edge device to perform DNSSEC validation directly
- Which is fine, but will this approach scale?

What can we say about a DNS environment where every edge device that poses DNS queries performs their own DNSSEC validation?

DNSSEC today

- A small, but growing, fraction of all domain names are signed using DNSSEC
- A larger, but still small, fraction of users use DNS resolvers that perform DNSSEC validation



At the end of August 2014, some ~11.5% of users send their DNS queries to DNSSEC-validating resolvers

What if everyone did it?

What if:

every resolver performed DNSSEC validation?

or even if:

every end device performed DNSSEC validation?

What difference in traffic loads and query rates would we see at an authoritative name server between serving an unsigned domain name and serving the signed equivalent of the domain name?

The Experiment

- We serve an online Ad with 3 embedded URLs that the user's browser is tasked to fetch. The URLs use unique domain names that are:
  - Unsigned
  - Signed (good)
  - Signed (bad)

- We are looking for behaviours where we see the browser perform:
  - Queries for the DS and DNSKEY RRs for both of the the signed domains, and
  - Fetch the signed (good) but not the signed (bad) URLs

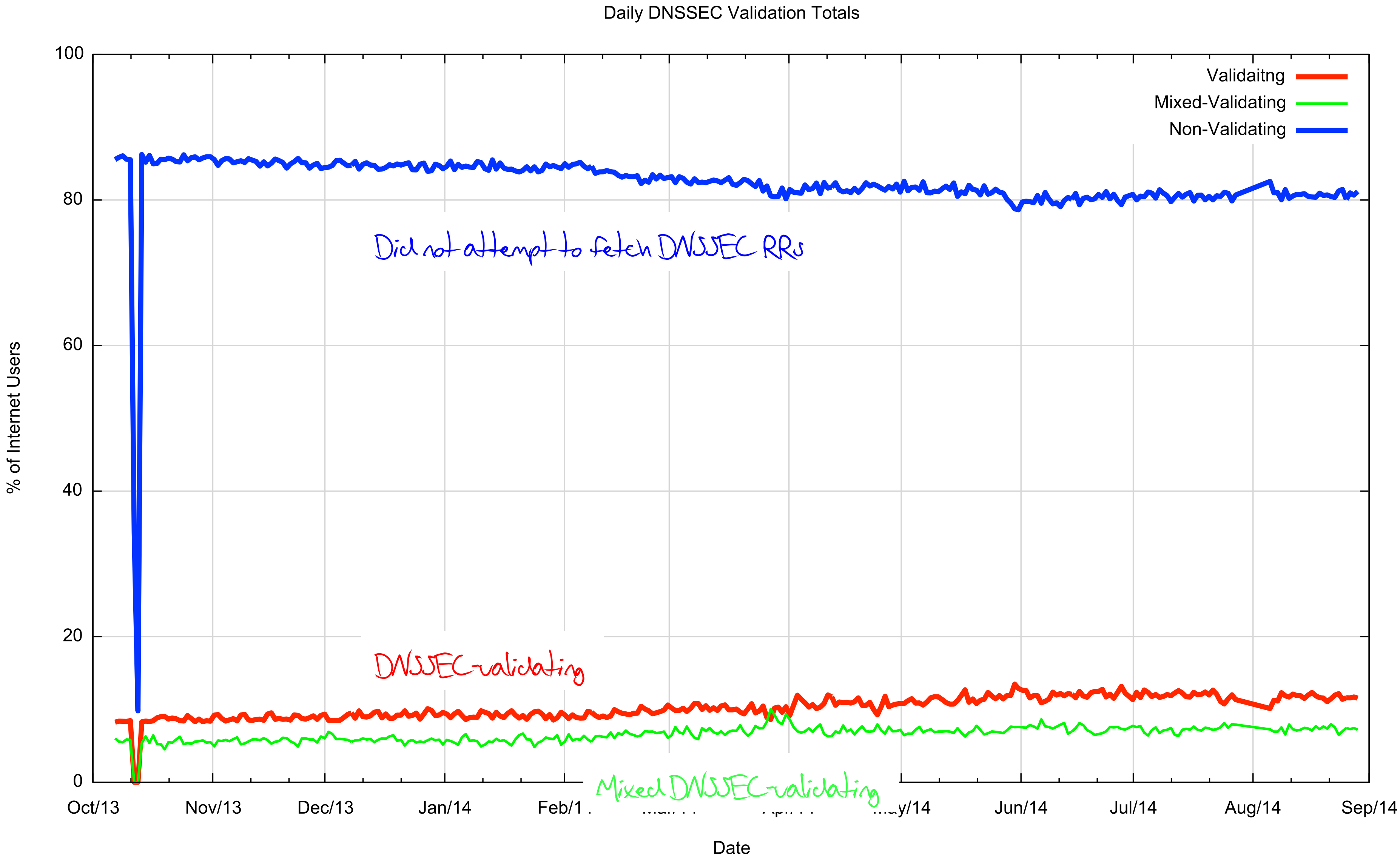
What we saw

- Users who exclusively used DNSSEC-validating resolvers
- Users who used a mix of validating and non-validating resolvers

  (typically, we saw the SERVFAIL response on a badly signed domain name cause the user to repeat the query to a resolver that did not perform DNSSEC validation)

- Users who exclusively used non-validating resolvers

What we saw



Did not attempt to fetch DNSSEC RRs

DNSSEC-validating

Mixed DNSSEC-validating

If your resolver validates DNS responses…

- Then the resolver will need to fetch the DNSKEY and DS RRs for the zone, and recurse upward to the root
- If these RRs are not cached, then at a minimum there are at least two additional DNS queries that are performed as part of the validation process

If your resolver validates DNS responses…

More queries, longer resolution time

Dual Stack client - query for unsigned domain name

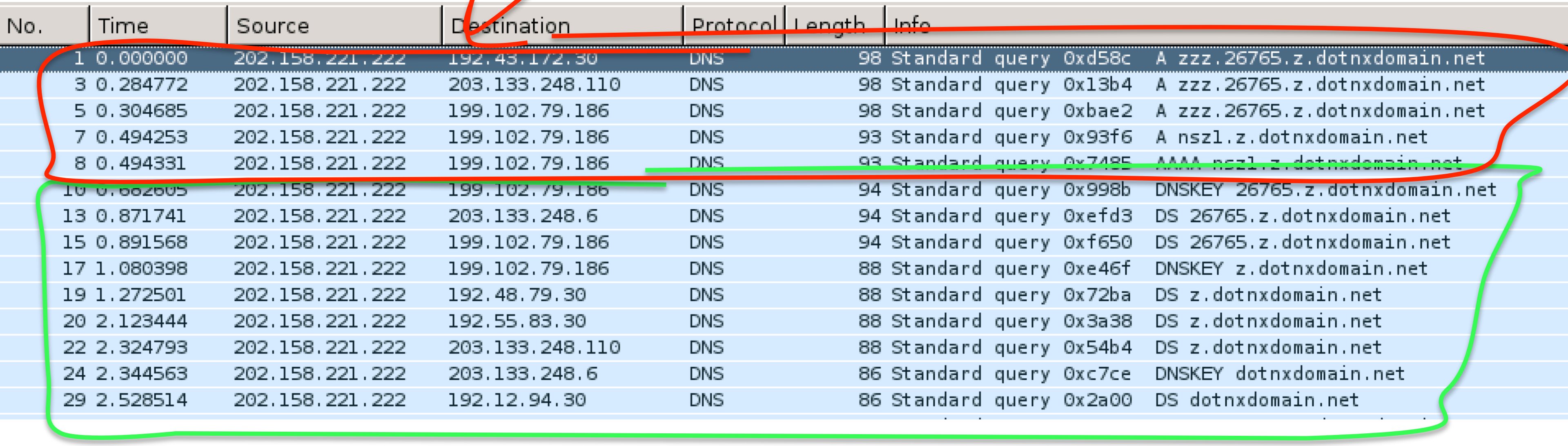
```
20:36:40.288 query: unsigned.example.com IN AAAA -ED (199.102.79.186)
20:36:41.028 query: unsigned.example.com IN A    -ED (199.102.79.186)
```

Dual Stack client - query for signed domain name

```
20:36:41.749 query: signed.example.com IN A      -ED (199.102.79.186)
20:36:41.758 query: signed.example.com IN AAAA   -ED (199.102.79.186)
20:36:41.876 query: signed.example.com IN DS     -ED (199.102.79.186)
20:36:41.993 query: signed.example.com IN DNSKEY -ED (199.102.79.186)
```

Validation - DNS Queries

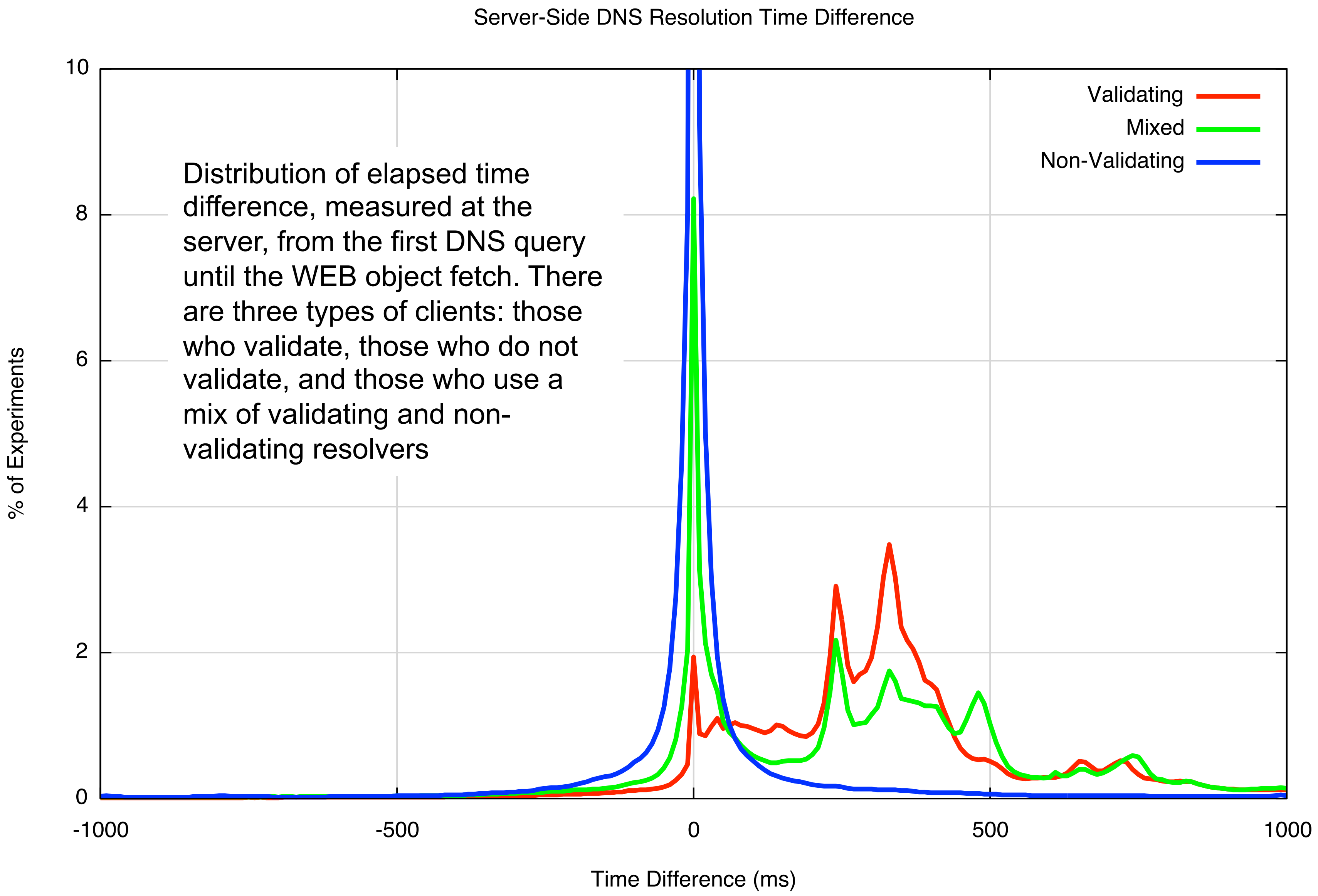
DNS queries



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	202.158.221.222	192.43.172.30	DNS	98	Standard query 0xd58c A zzz.26765.z.dotnxdomain.net
3	0.284772	202.158.221.222	203.133.248.110	DNS	98	Standard query 0x13b4 A zzz.26765.z.dotnxdomain.net
5	0.304685	202.158.221.222	199.102.79.186	DNS	98	Standard query 0xbae2 A zzz.26765.z.dotnxdomain.net
7	0.494253	202.158.221.222	199.102.79.186	DNS	93	Standard query 0x93f6 A nsz1.z.dotnxdomain.net
8	0.494331	202.158.221.222	199.102.79.186	DNS	93	Standard query 0x7485 AAAA nsz1.z.dotnxdomain.net
10	0.682605	202.158.221.222	199.102.79.186	DNS	94	Standard query 0x998b DNSKEY 26765.z.dotnxdomain.net
13	0.871741	202.158.221.222	203.133.248.6	DNS	94	Standard query 0xefd3 DS 26765.z.dotnxdomain.net
15	0.891568	202.158.221.222	199.102.79.186	DNS	94	Standard query 0xf650 DS 26765.z.dotnxdomain.net
17	1.080398	202.158.221.222	199.102.79.186	DNS	88	Standard query 0xe46f DNSKEY z.dotnxdomain.net
19	1.272501	202.158.221.222	192.48.79.30	DNS	88	Standard query 0x72ba DS z.dotnxdomain.net
20	2.123444	202.158.221.222	192.55.83.30	DNS	88	Standard query 0x3a38 DS z.dotnxdomain.net
22	2.324793	202.158.221.222	203.133.248.110	DNS	88	Standard query 0x54b4 DS z.dotnxdomain.net
24	2.344563	202.158.221.222	203.133.248.6	DNS	86	Standard query 0xc7ce DNSKEY dotnxdomain.net
29	2.528514	202.158.221.222	192.12.94.30	DNS	86	Standard query 0x2a00 DS dotnxdomain.net

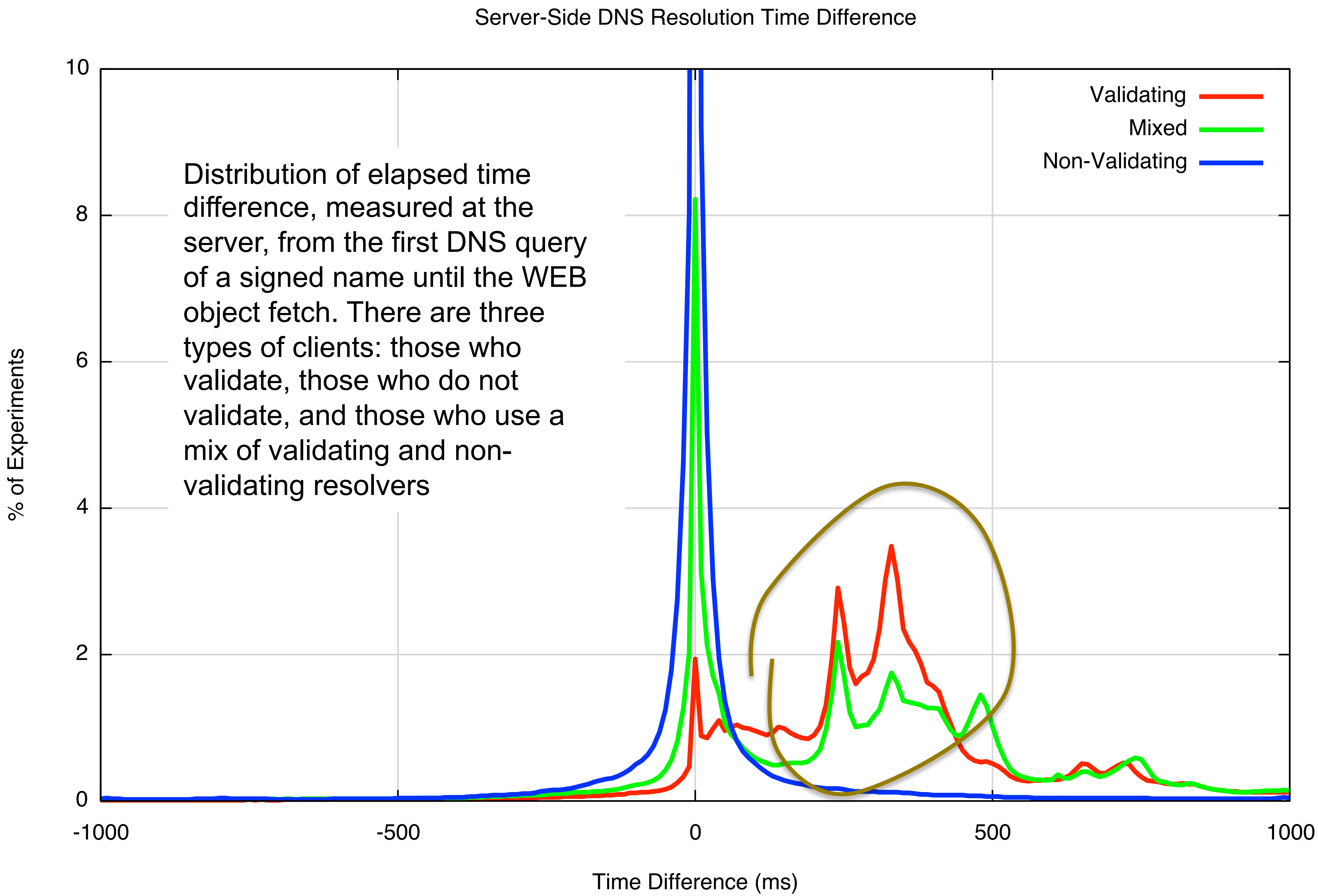
Validation Queries

Measured Time Cost



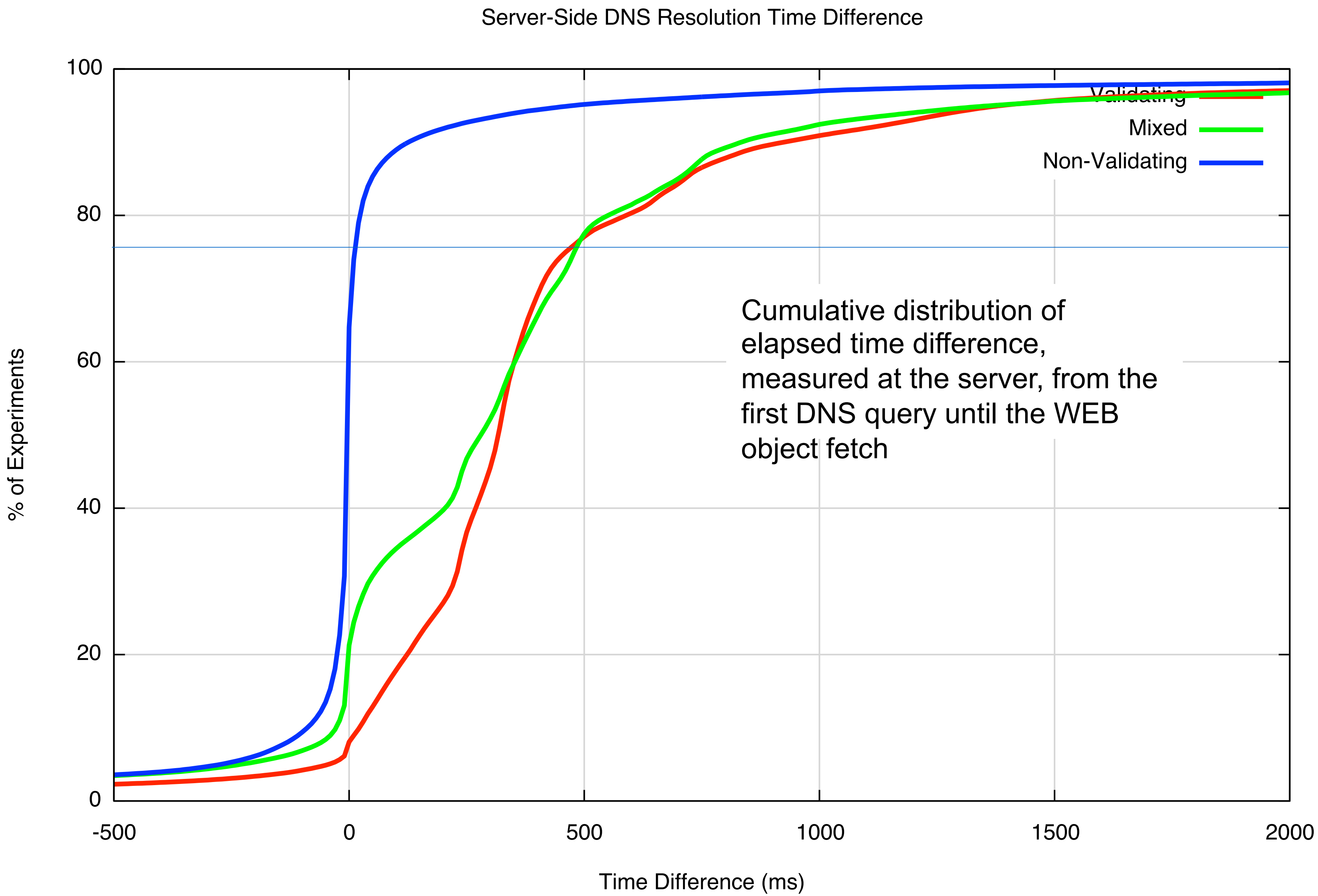
Distribution of elapsed time difference, measured at the server, from the first DNS query until the WEB object fetch. There are three types of clients: those who validate, those who do not validate, and those who use a mix of validating and non-validating resolvers

Measured Time Cost



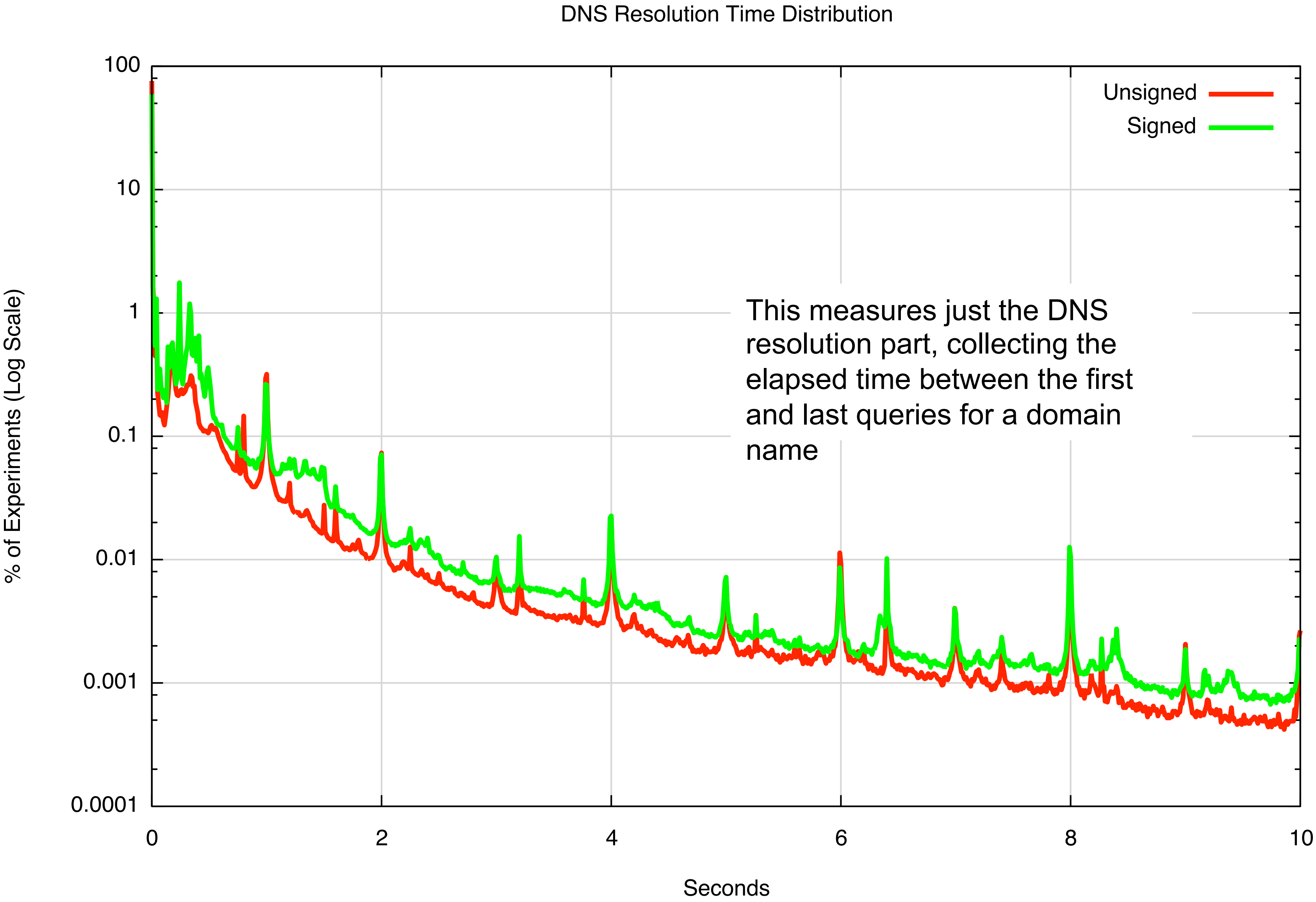
Distribution of elapsed time difference, measured at the server, from the first DNS query of a signed name until the WEB object fetch. There are three types of clients: those who validate, those who do not validate, and those who use a mix of validating and non-validating resolvers

Time Cost



Cumulative distribution of elapsed time difference, measured at the server, from the first DNS query until the WEB object fetch

DNS Resolution Time



This measures just the DNS resolution part, collecting the elapsed time between the first and last queries for a domain name

Unsigned/Non-Validating vs Signed/Validating

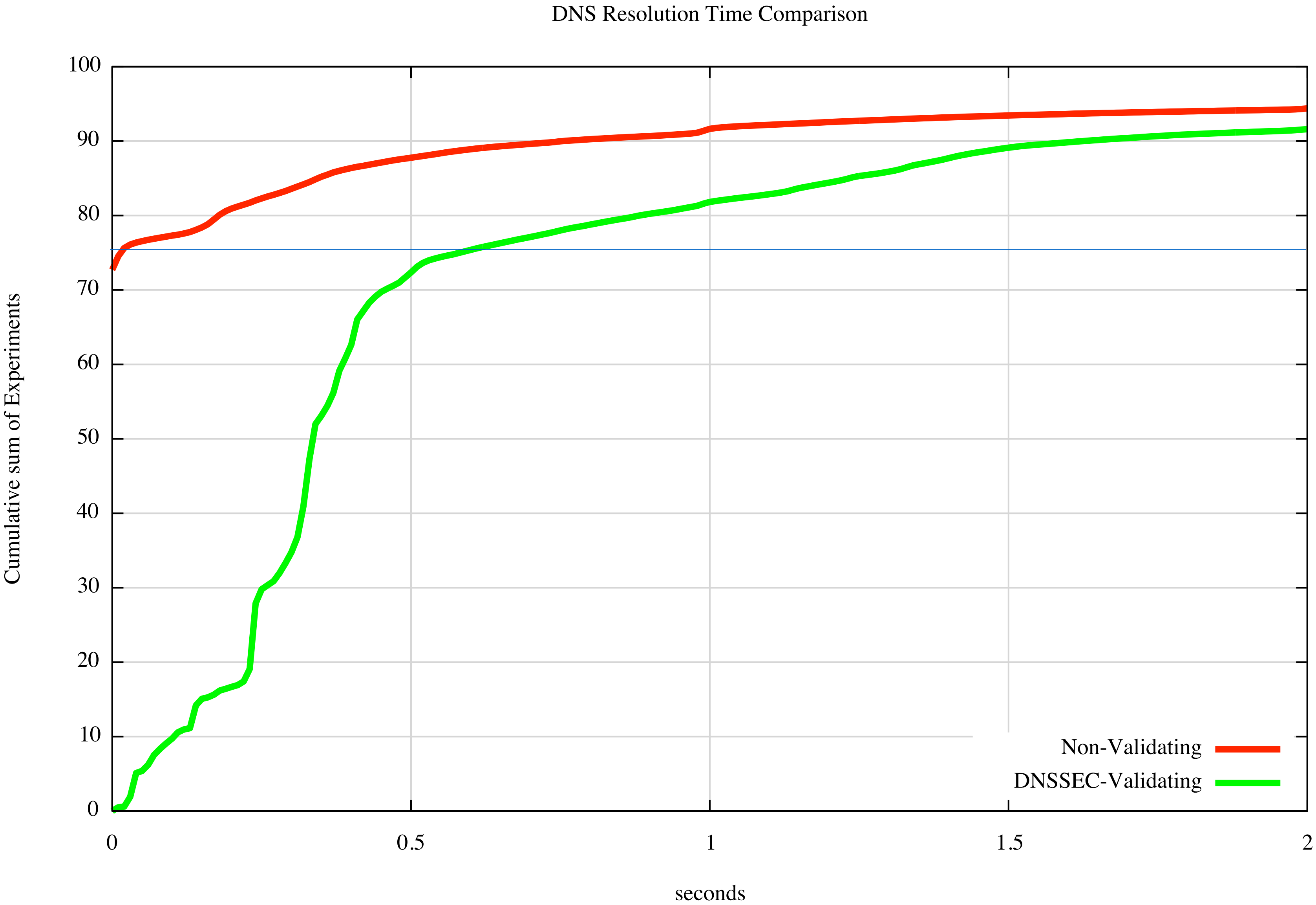
Let's try a slightly different comparison, and compare the total DNS query time between

- Non-validating users querying an unsigned name

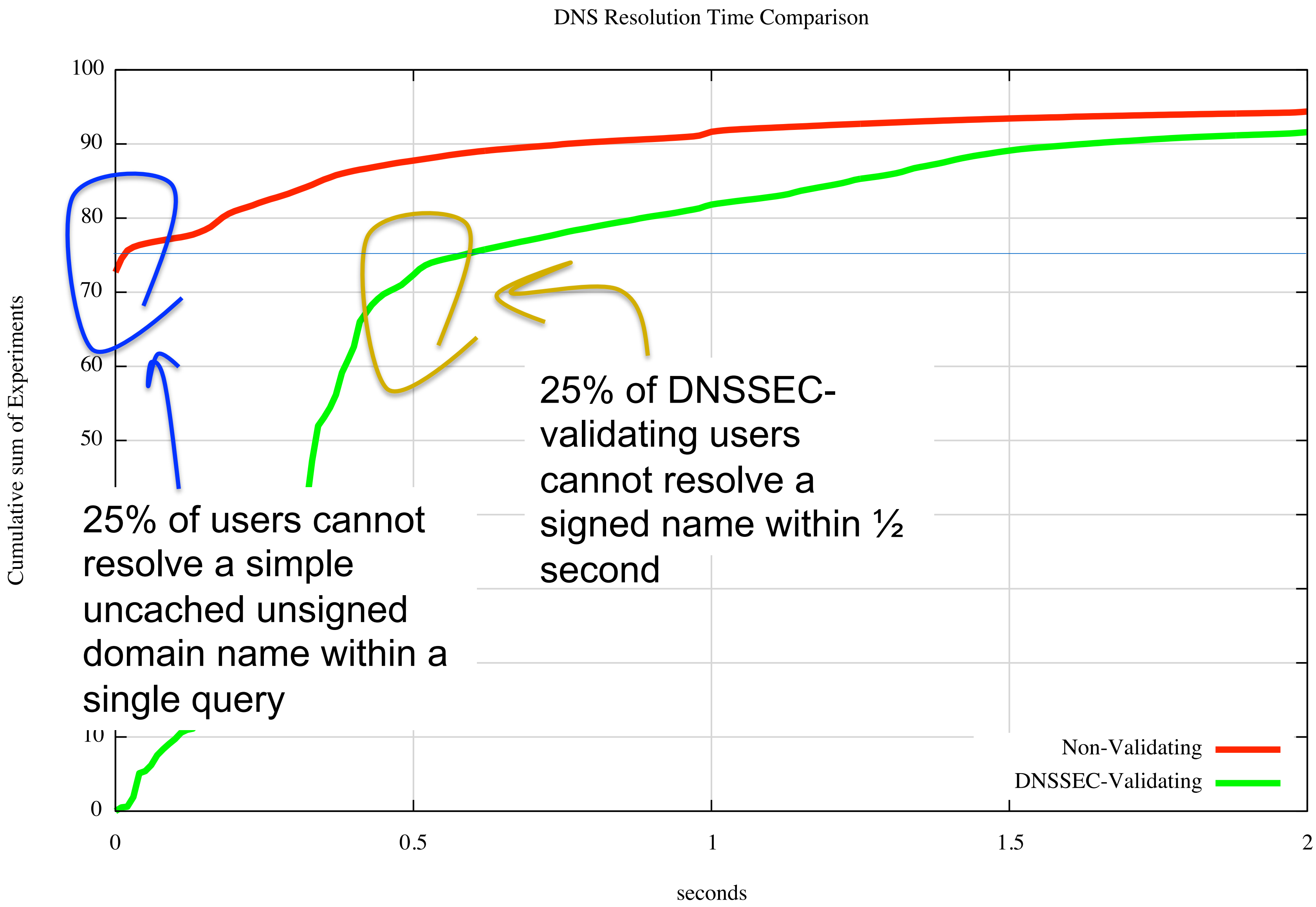
  and

- Validating users querying for a signed name

Like-vs-like: unsigned vs signed



Like-vs-like: unsigned vs signed



25% of users cannot resolve a simple uncached unsigned domain name within a single query

25% of DNSSEC-validating users cannot resolve a signed name within ½ second

Validation Time

- When resolving a previously unseen domain name most clients will experience up to 500ms additional time spent in validation
  - This is due to the additional queries related to the fetch of the DNSKEY / DS RR sequence to validate the RRSIG of the original response

This validation phase could be processed in less time…

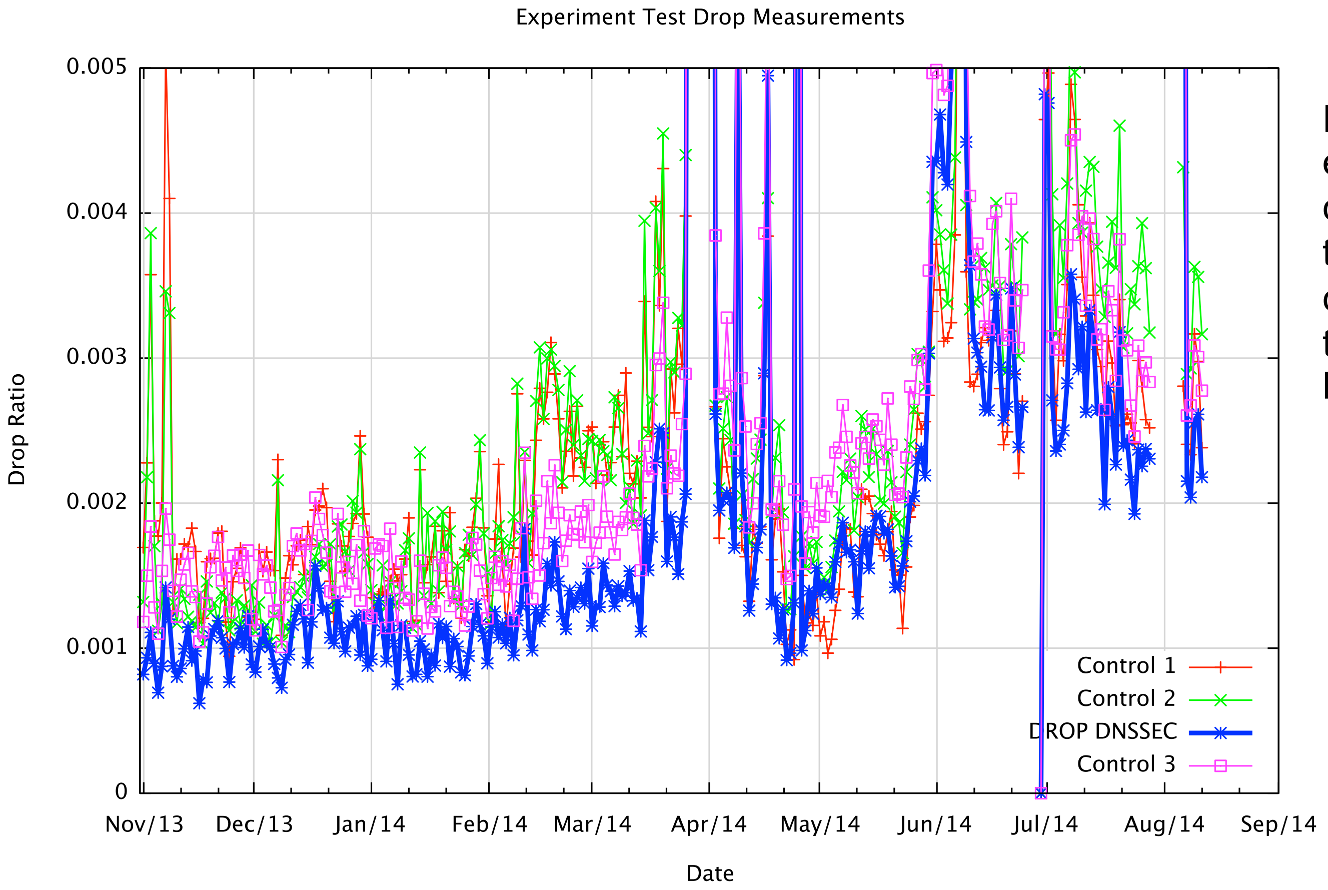
- Most resolvers appear to perform the validation path check using serial fetches. Parallel fetches of the DNSSEC validation path RRs would improve this situation so that the validation fetches would take a single query cycle time

Do any clients drop out?

Does the addition of the DNSSEC RR’s in the response cause any clients to stop attempts at DNS resolution?

So we looked…

Do any clients drop out?



If there was any clear evidence of DNSSEC causing resolution failure then the blue line would be clearly higher than the other three control lines
But its not.

There is no experimental evidence to suggest systematic resolution failure here for DNSSEC-signed names
However, the DNS responses in this experiment were all below 1500 octets.
We have yet to test the case of forced UDP fragmentation in DNS responses

Client Behaviour

- Retrieving DNSSEC credentials takes additional time and volume when validating the resolution outcomes of a signed name
- But much of this overhead is mitigated by use of local caches in the DNS resolution path
- And if resolvers performed validation using parallel fetches, the additional overhead could be brought down to a single retrieval cycle time

Authoritative Server Measurements

The following analysis attempts to answer the question:

- What increase in queries and traffic should I expect to see if the unsigned zone I currently serve is DNSSEC signed, and everyone is using DNSSEC validating resolvers?

If you serve a signed Domain Name:

You will generate larger responses:

Dual Stack client – query for unsigned domain name, no EDNS0

Query: 117 Bytes
Response: 168 bytes

Dual Stack client – query for signed domain name, EDNS0

Query: (A) 127 Bytes
Response: (A) 1168 bytes

Query: (DS) 80 Bytes
Response: (DS) 341 bytes

Query: (DNSKEY) 80 Bytes
Response: (DNSKEY) 742 bytes

Total: Query: 287 bytes
Response: 2,251 bytes

If you serve a signed Domain Name:

You will generate larger responses:

Dual Stack client – query for unsigned domain name, no EDNS0

Query: 117 Bytes
Response: 168 bytes

Dual Stack client – query for signed domain name, EDNS0

Query: (A) 127 Bytes
Response: (A) 1168 bytes

Query: (DS) 80 Bytes
Response: (DS) 341 bytes

Query: (DNSKEY) 80 Bytes
Response: (DNSKEY) 742 byte

Total: Query: 287 byte
Response: 2,251 bytes

The DS query is directed to the parent zone, so you may or may not see this query at the authoritative server. In our case we are serving the parent zone as well

If you serve a signed Domain Name:

You will generate larger responses:

Dual Stack client – query for unsigned domain name, no EDNS0

Query: 117 Bytes
Response: 168 bytes

Dual Stack client – query for signed domain name, EDNS0

Query: (A) 127 Bytes
Response: (A) 1168 byt

Query: (DS) 80 Bytes
Response: (DS) 341 bytes

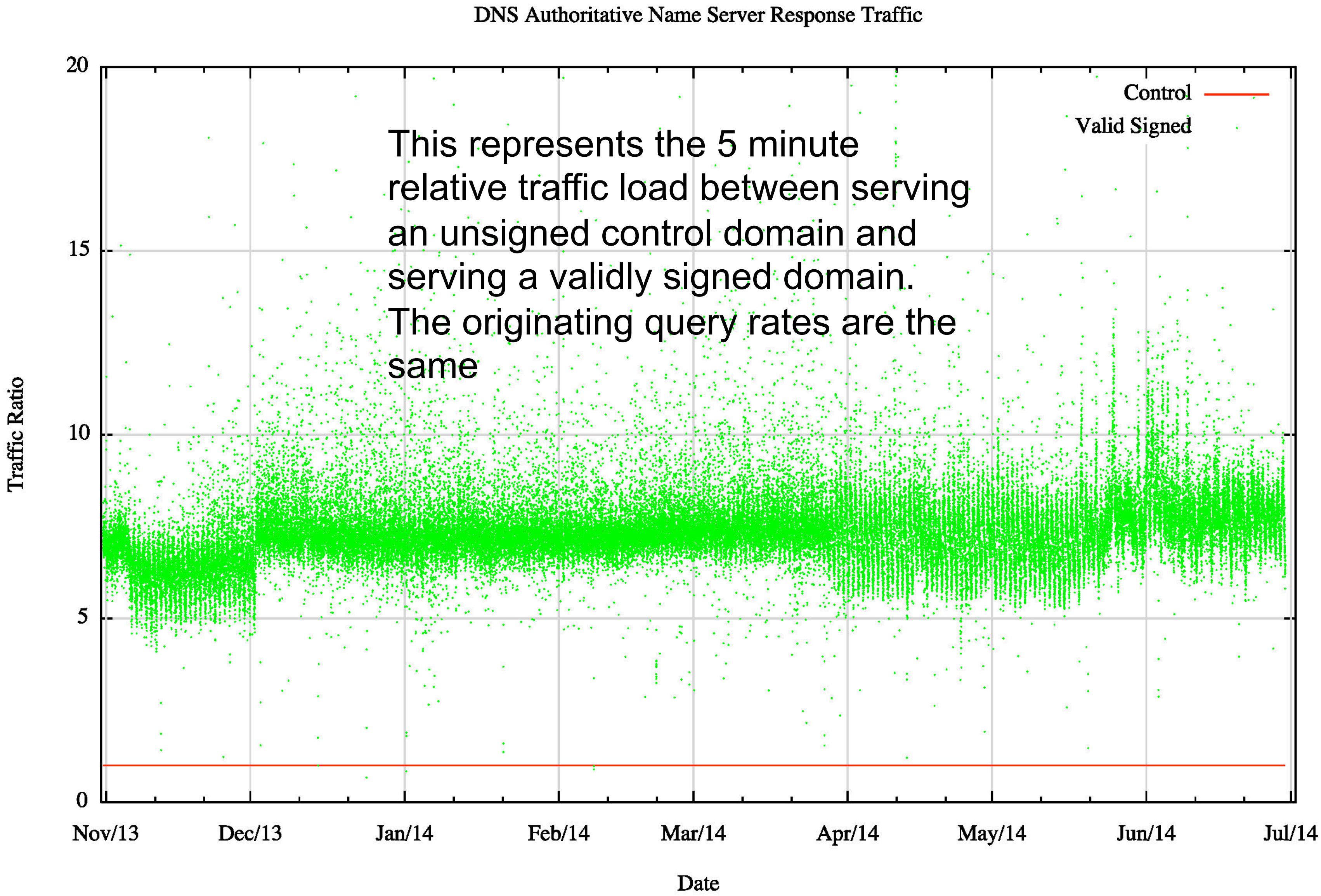
Query: (DNSKEY) 80 Bytes
Response: (DNSKEY) 742 byte

Total: Query: 287 bytes
Response: 2,251 bytes

That's an increase of 13x in terms of outbound traffic volume

The DS query is directed to the parent zone, so you may or may not see this query at the authoritative server. In our case we are serving the parent zone as well

Server Traffic Load



This represents the 5 minute relative traffic load between serving an unsigned control domain and serving a validly signed domain. The originating query rates are the same

Server Traffic Load

- Serving a DNSSEC-signed name appears to generate 7.5x the traffic load, as compared to serving an unsigned name
  - But 20% of clients are performing validation, and hence 20% of the clients generate 13x more traffic
  - The theory would expect to see a 3.4x increase in traffic.
  - Why is this observed result double the prediction?

Server Traffic Load

- Use of the EDNS DNSSEC-OK flag is far higher than the level of DNSSEC validation
  - 84% of queries have the EDNS0 DNSSEC-OK flag set
  - And this query generates a response of 1168 bytes (i.e. 7x the size of a null EDNS response)
  - So 64% of clients set EDNS0 DNSSEC-OK, and 20% of clients also ask for DS and DNSKEY RRs
  - The theory predicts that this would result in 7.25x the traffic over an unsigned domain
  - Which is (roughly) what we see

Server Traffic Load

- What is the traffic load difference between serving an unsigned zone and serving a signed zone if **every** client performed DNSSEC validation?
  - The difference from the current levels of DNSSEC traffic lies predominately in the additional DNSKEY and DS responses
  - You should expect approximately **15x** the traffic load for response traffic

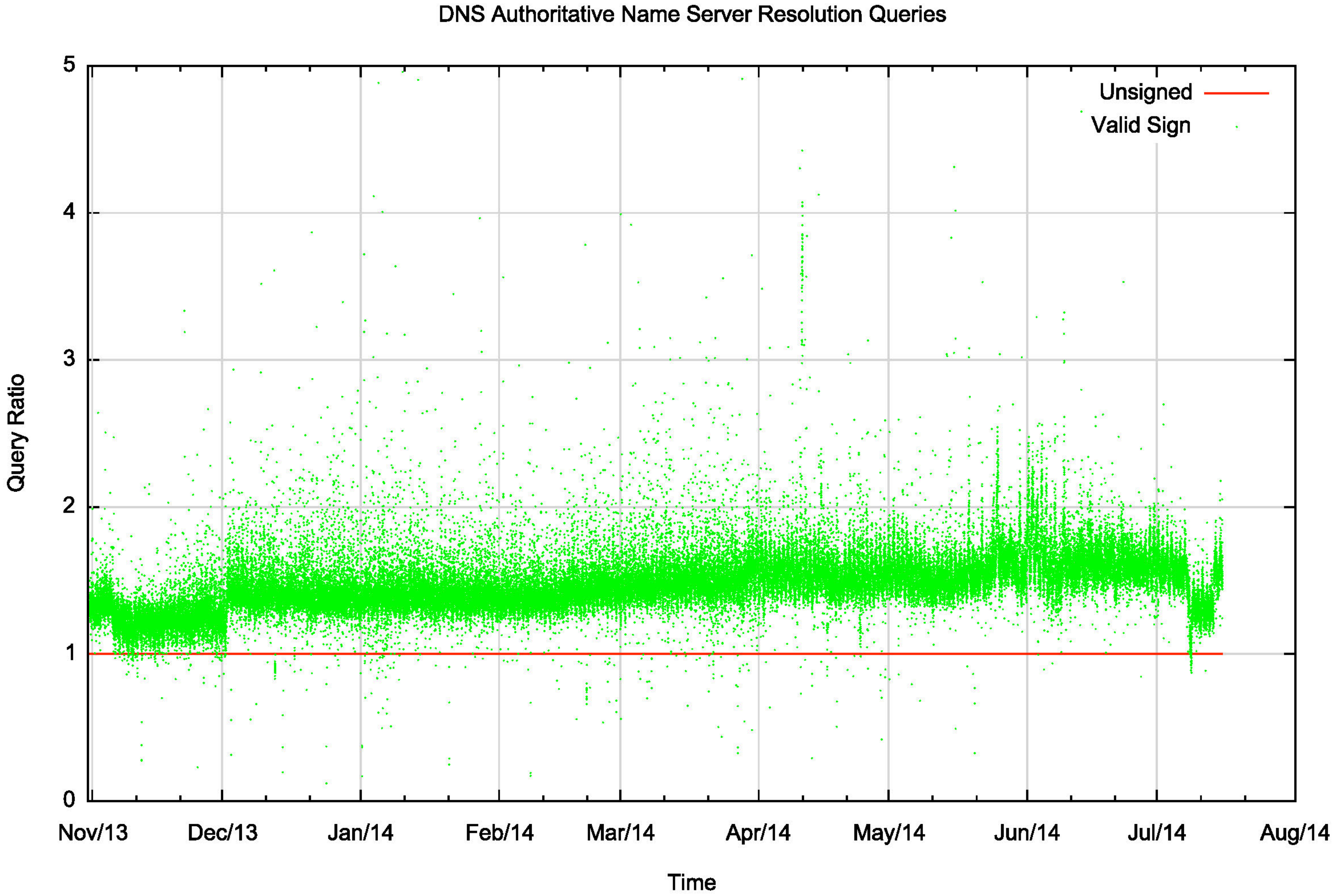
If you serve a signed Domain Name:

You'll receive 2-3 times as many queries:

```
    Dual Stack client – query for unsigned domain name, no EDNS0

            Query: 117 Bytes
            Response: 168 bytes

    Dual Stack client – query for signed domain name

            Query: (A) 127 Bytes
            Response: (A) 1168 byte

            Query: (DS) 80 Bytes
            Response: (DS) 341 bytes

            Query: (DNSKEY) 80 Bytes
            Response: (DNSKEY) 742 by
```

The DS query is directed to the parent zone, so you may or may not see this query at the authoritative server. In our case we are serving the parent zone as well

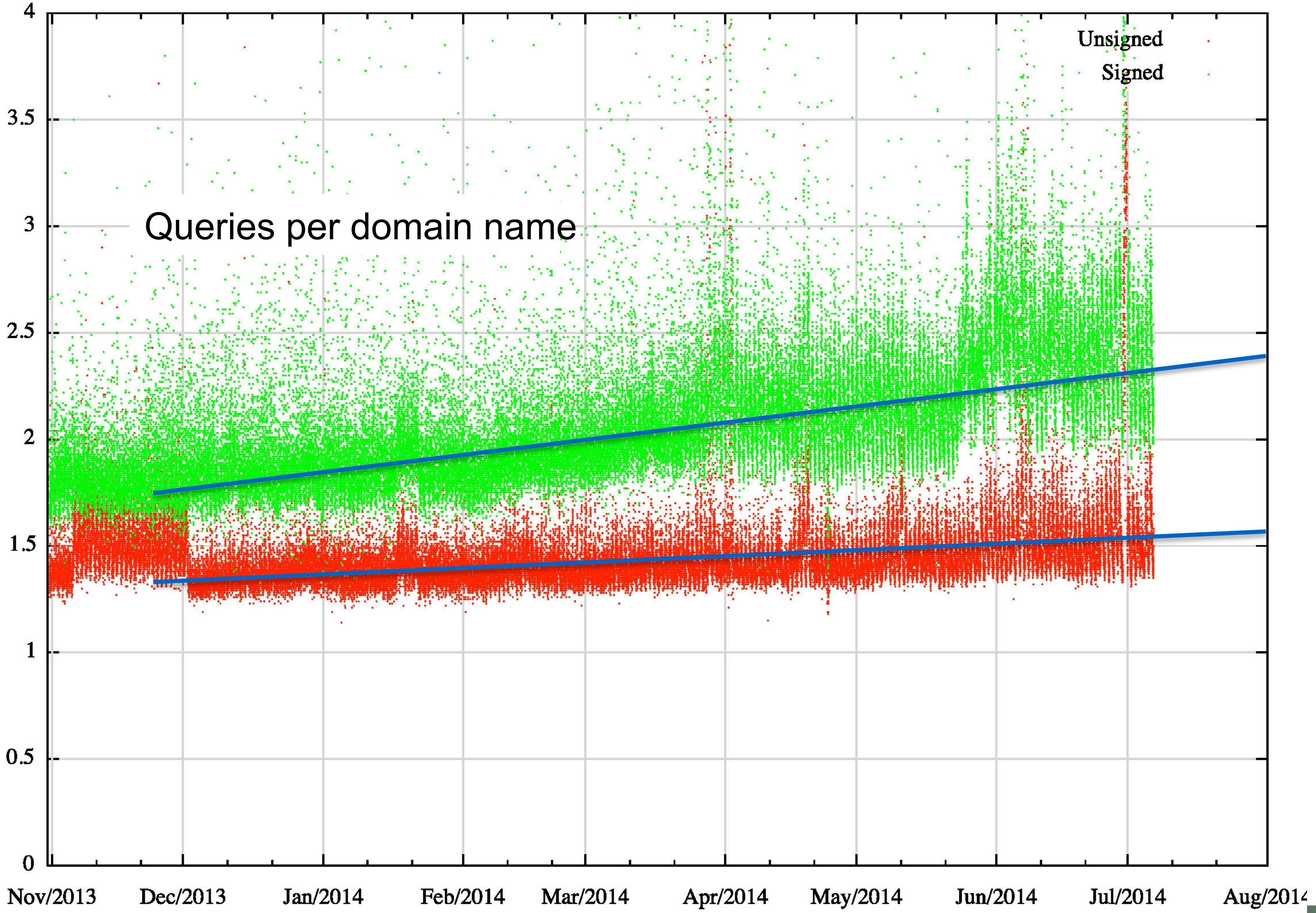
Server Query Load



Server Query Load

- 20% of clients use validating resolvers, so the signed domain query load should be 1.4x that of the unsigned domain
- But we are observing an increase in the query load of 1.6x the unsigned domain.
- Why?

Repeat queries are rising



Queries per domain name

Query duplication

We are seeing a noticeable level of query duplication from anycast DNS server farms

The same query is being received from multiple slave resolvers within a short period of time

```
Domain             Time         Query source               Query

0a62f.z.example.com 02:05:31.998 74.125.41.81  port: 52065  q: DNSKEY?
0a62f.z.example.com 02:05:32.000 74.125.41.19  port: 53887  q: DNSKEY?
0a62f.z.example.com 02:05:32.005 74.125.41.146 port: 52189  q: DNSKEY?
0a62f.z.example.com 02:05:32.008 74.125.16.213 port: 42079  q: DNSKEY?
```

This is rising over time

Setting Expectations

For a validly signed zone an authoritative server may anticipate about **4x the query load** and **15x the traffic load** as compared to serving an equivalent unsigned zone, if everyone performed DNSSEC validation *

(* if you served the parent zone as well)

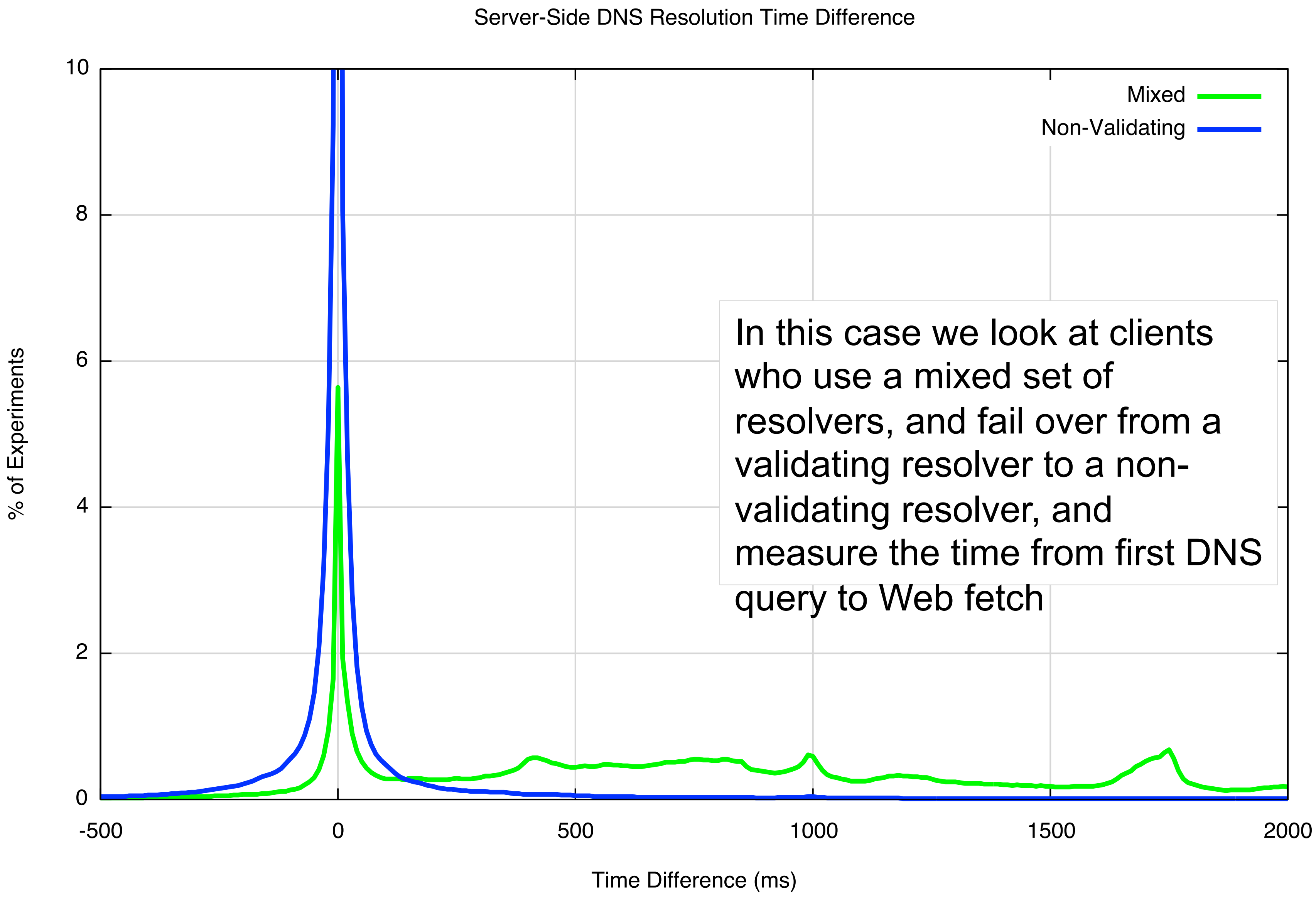
The Worst Case

But things get worse when the DNSSEC signatures are invalid:

- The response from a DNSSEC-validating recursive resolver upon DNSSEC validation failure is SERVFAIL, which prompts clients of this resolver to re-query using an alternative resolver
- The recursive resolver may re-query the name using alternative servers, on the assumption that the validation failure is due to a secondary server falling out of sync with the current zone data

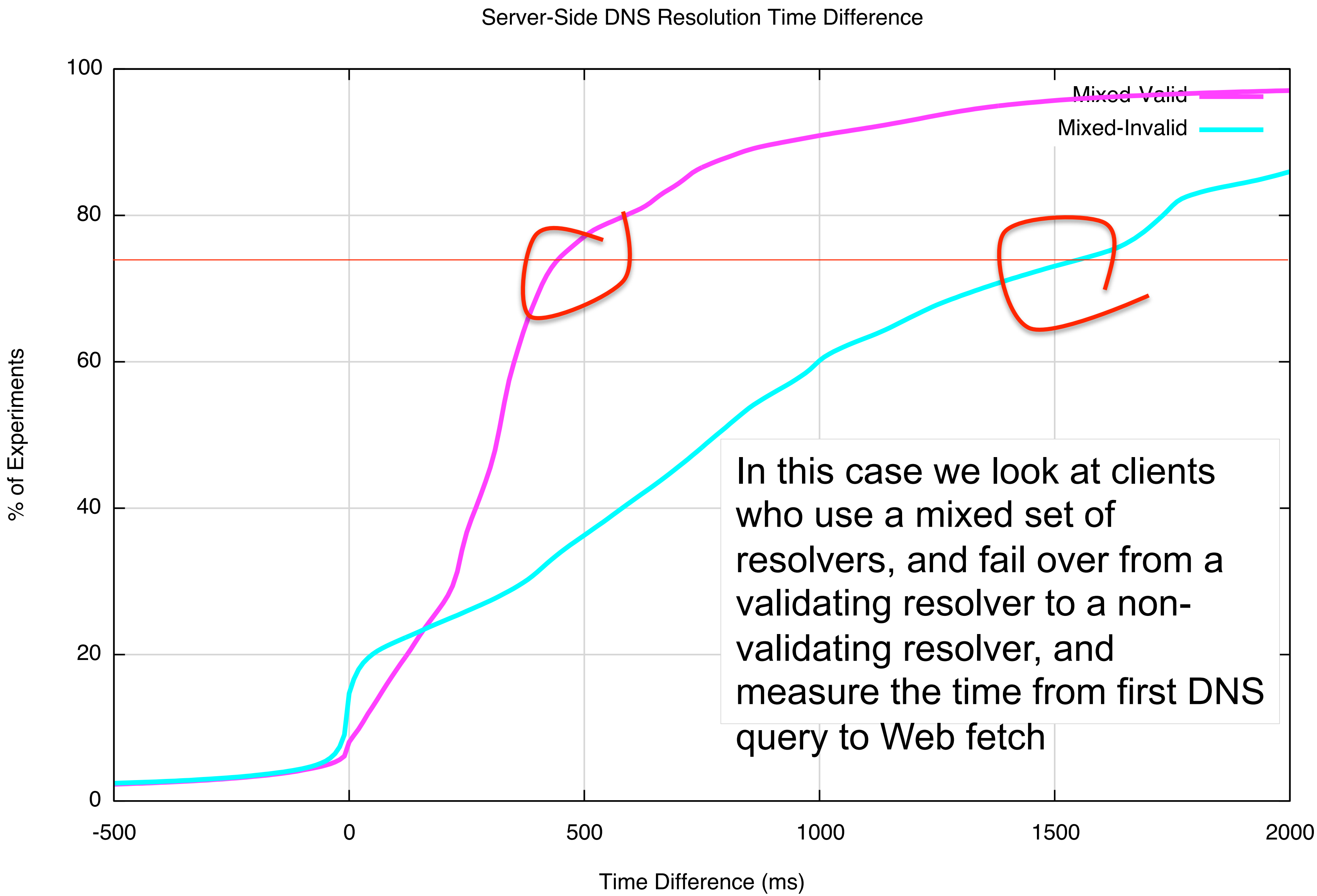
How much worse does it get?

DNS Resolution Time Difference



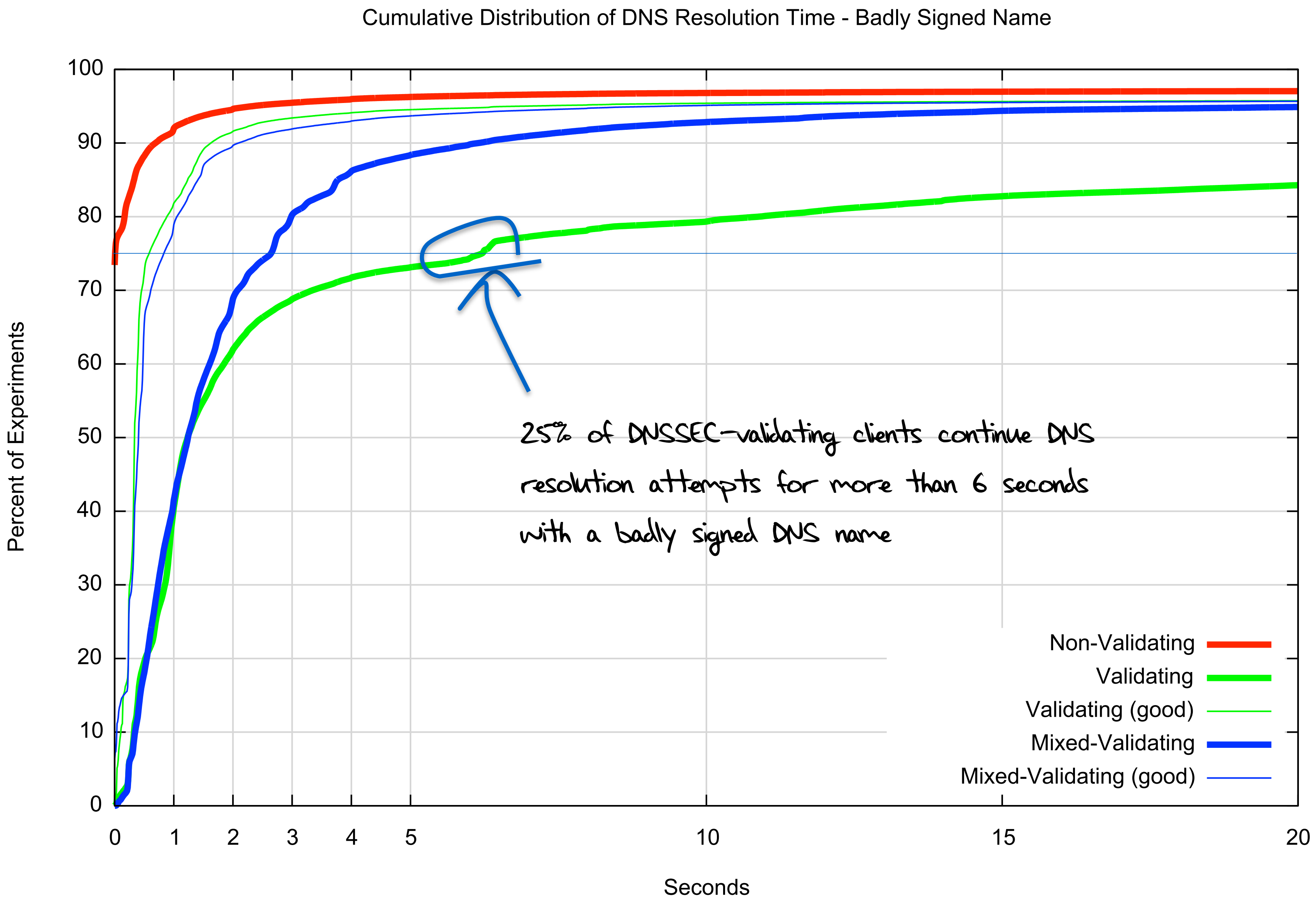
In this case we look at clients who use a mixed set of resolvers, and fail over from a validating resolver to a non-validating resolver, and measure the time from first DNS query to Web fetch

DNS Resolution Time Difference



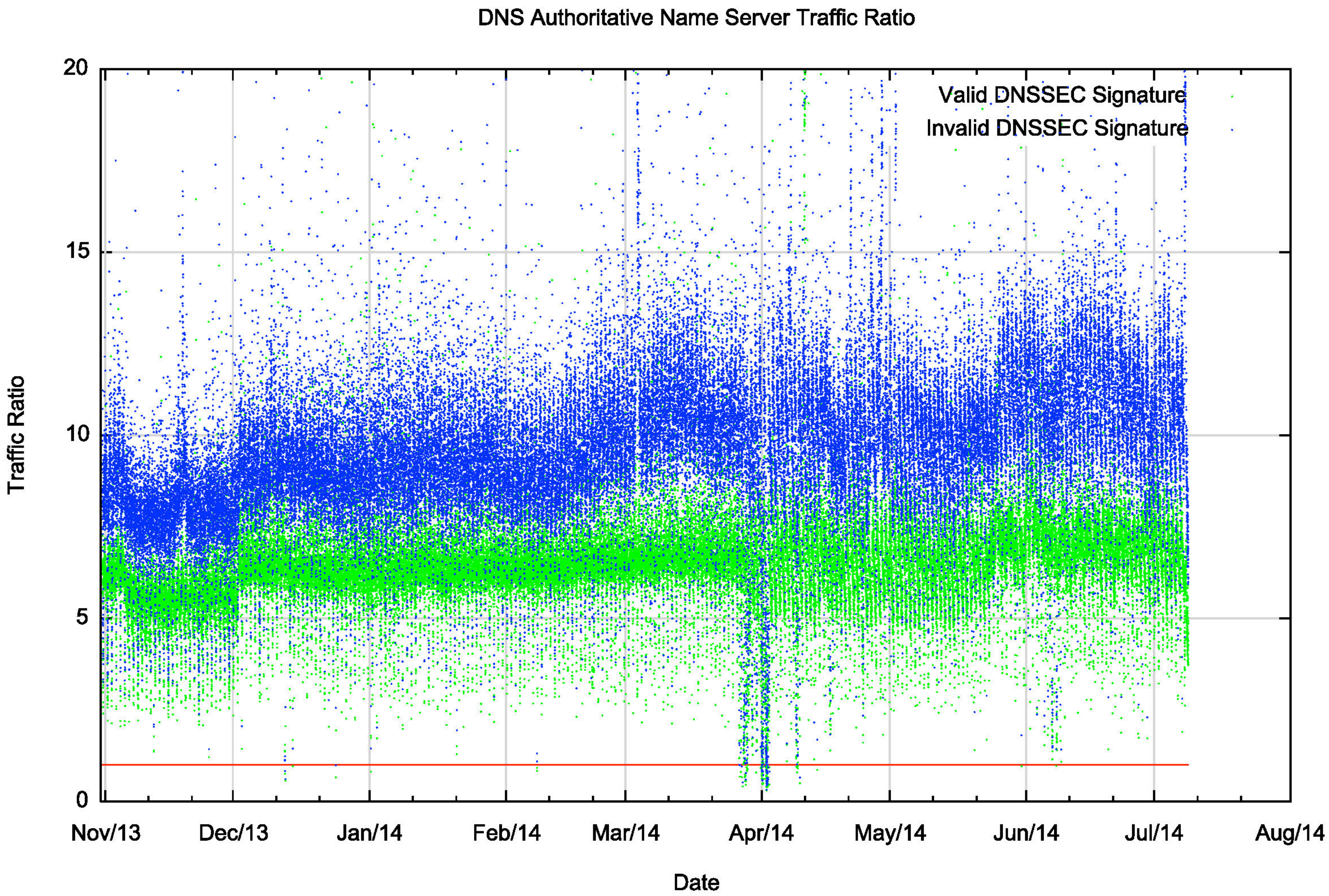
In this case we look at clients who use a mixed set of resolvers, and fail over from a validating resolver to a non-validating resolver, and measure the time from first DNS query to Web fetch

DNS Resolution Times



25% of DNSSEC-validating clients continue DNS resolution attempts for more than 6 seconds with a badly signed DNS name

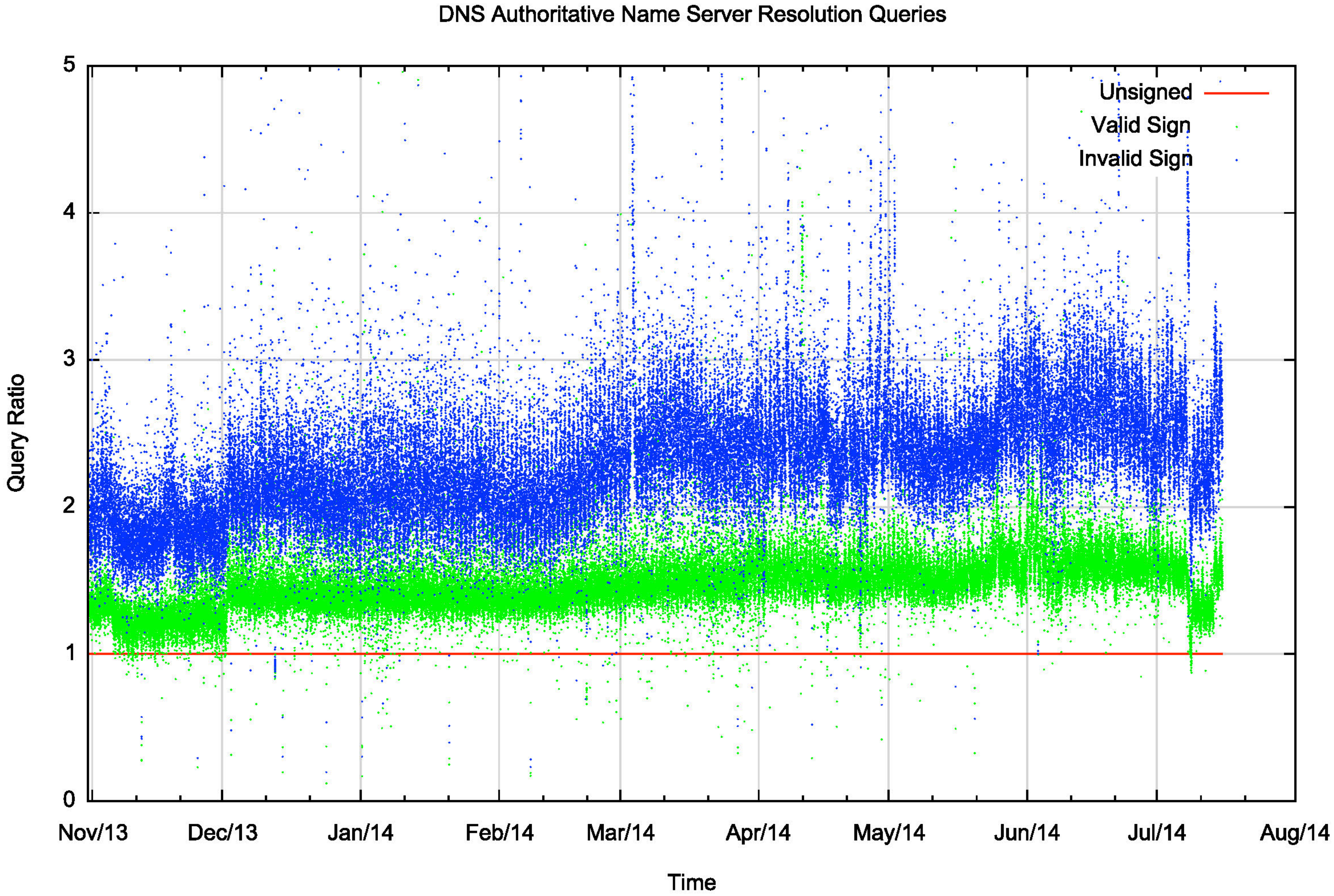
Relative Traffic Profile



Traffic Profile

- The traffic load for a badly signed domain name is around 10x the load for an unsigned domain
- If everyone were to use validating resolvers then the load profile would rise to around 26x the load of an unsigned domain

Query Profile



Query Profile

- The query load for a badly signed domain name is around 2.5x the load for an unsigned domain
- If everyone were to use validating resolvers then the load profile would rise to around 4x the load of an unsigned domain

Badly Signed Names

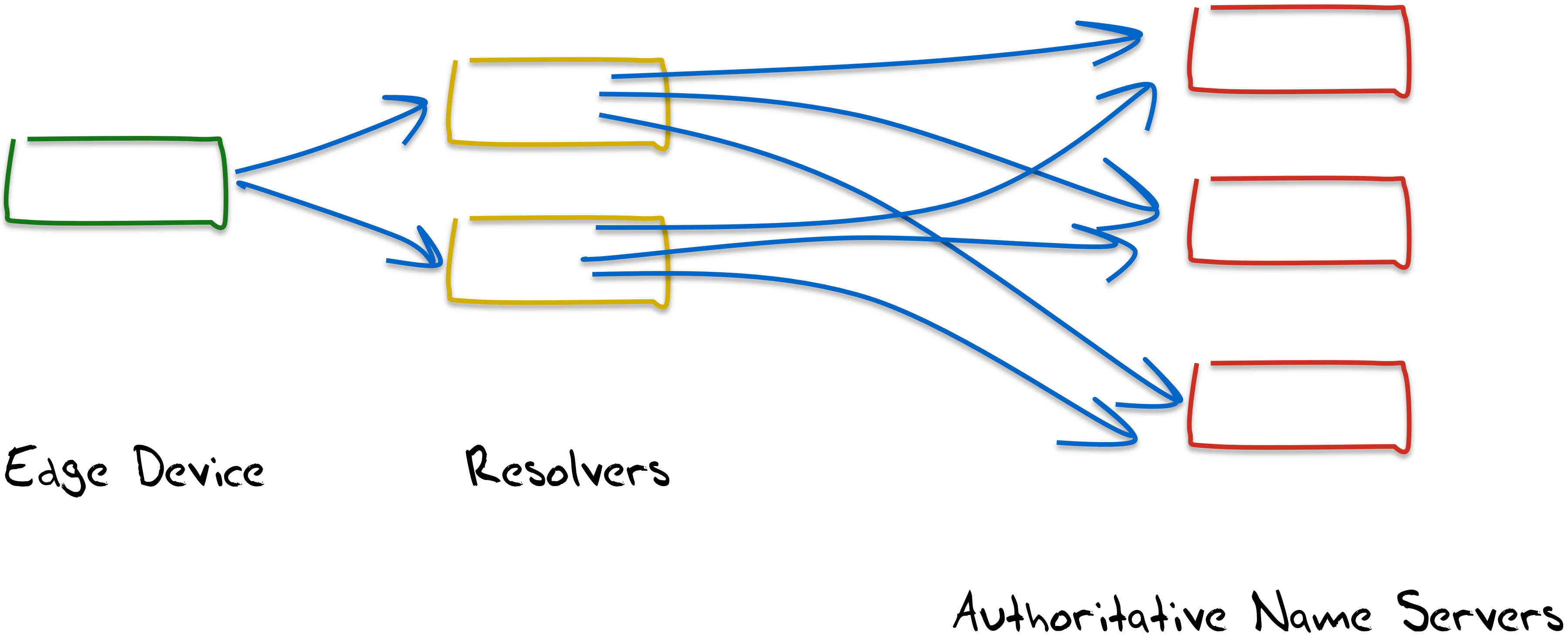
The problem with a badly signed name is the lack of caching – when a name does not validate, a validating resolver should not cache the resolution outcomes

So now all resolution attempts from validating resolvers generate queries at the authoritative name servers

And the use of a rather cryptic “ServFail” response prompts some recursive resolvers to query all nameservers

So the resultant query load on the authoritative name servers is far higher than these measurements would suggest

Badly Signed Names



Setting Expectations for DNSSEC

For a validly signed zone an authoritative server may anticipate about **4x the query load** and **15x the traffic load** as compared to serving an equivalent unsigned zone, if everyone performed DNSSEC validation *

But if you serve a badly signed zone, expect >>**8x the query load** and around >>**26x the traffic load** *

(* if you served the parent zone as well)

Thank You

Questions?