Using Long TTLs to Survive DNS Attacks

Duane Wessels

March 21, 2012

VERISIGN™

The Long TTL Proposal

- Use long TTLs on “infrastructure records” to help survive parent zone attacks and outages.
- NS RRs.
- A/AAAA RRs of name servers.
- Don’t forget DNSSEC: DS, DNSKEY, RRSIG, NSEC*?

- http://tools.ietf.org/html/draft-pappas-dnsop-long-ttl-04

Motivation

- There have been recent, publicized threats of attacks to root name servers.
    - So we think about and look at TTLs used by TLDs.
- These techniques work equally well to survive attacks against a parent zone.
- While the recent threats may not be credible, we nonetheless use this as an opportunity to explore ways of improving DNS resilience.

What Sort of Attacks Are We Talking About?

- Any attack where **all** the authoritative name servers for a given zone are non-responsive.
- If at least one name server for a zone is still responding, then TTLs may matter less.

VERISIGN™

You Don’t Need Me To Tell You This...

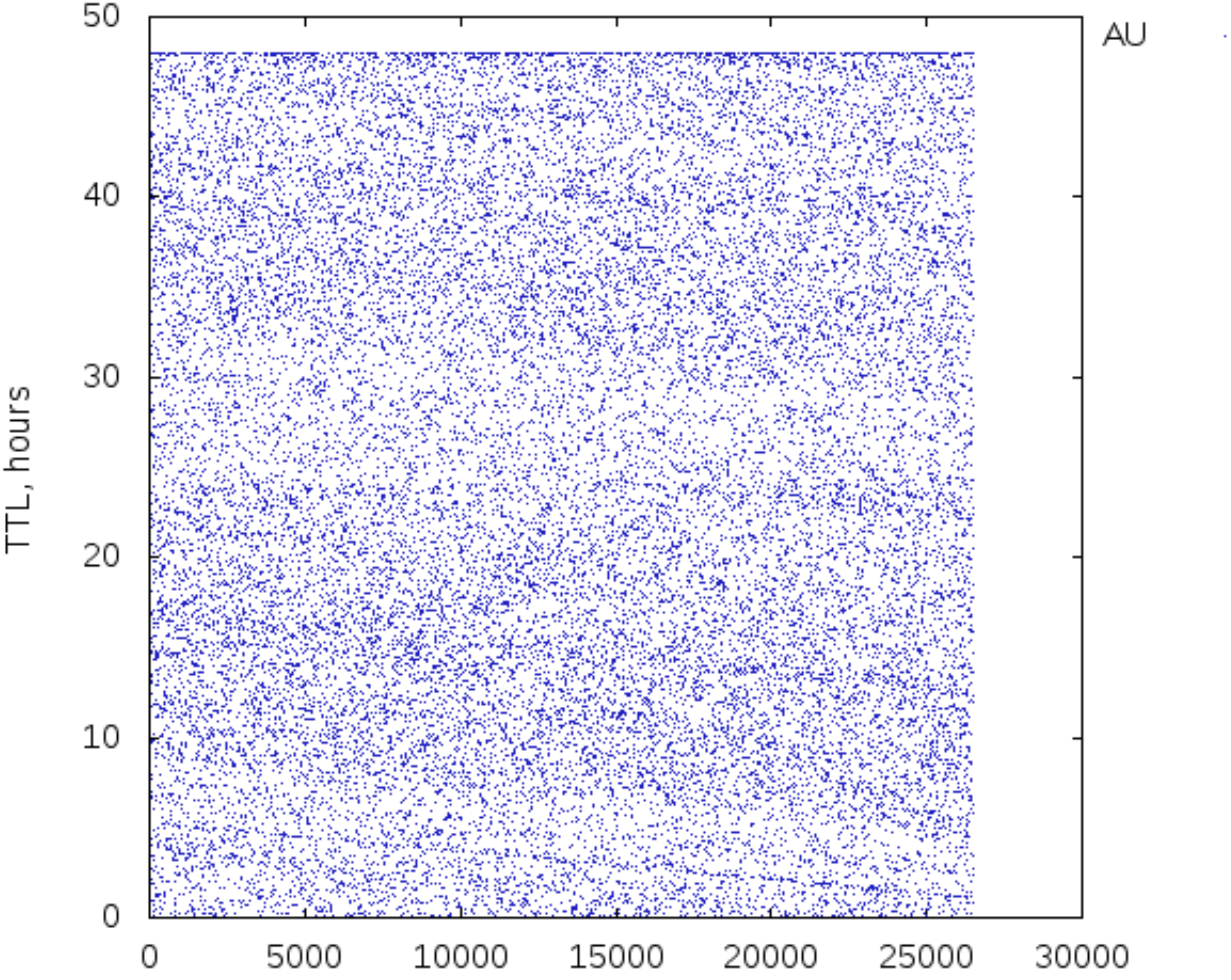
- Choice of TTLs is a tradeoff
- Higher TTLs
  - more stability
  - less flexibility
  - less traffic
- Lower TTLs
  - less stability
  - more flexibility
  - more traffic

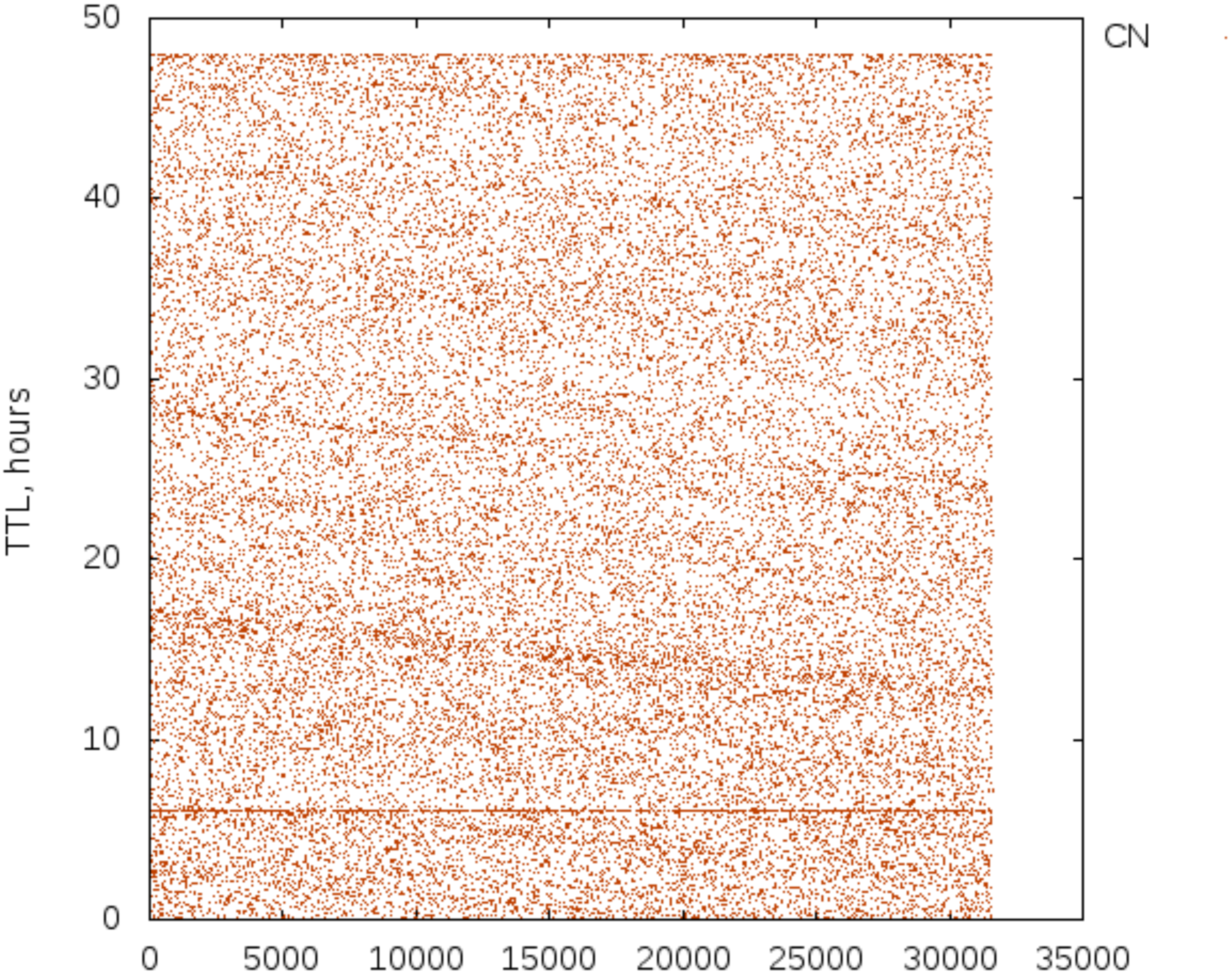
VERISIGN™

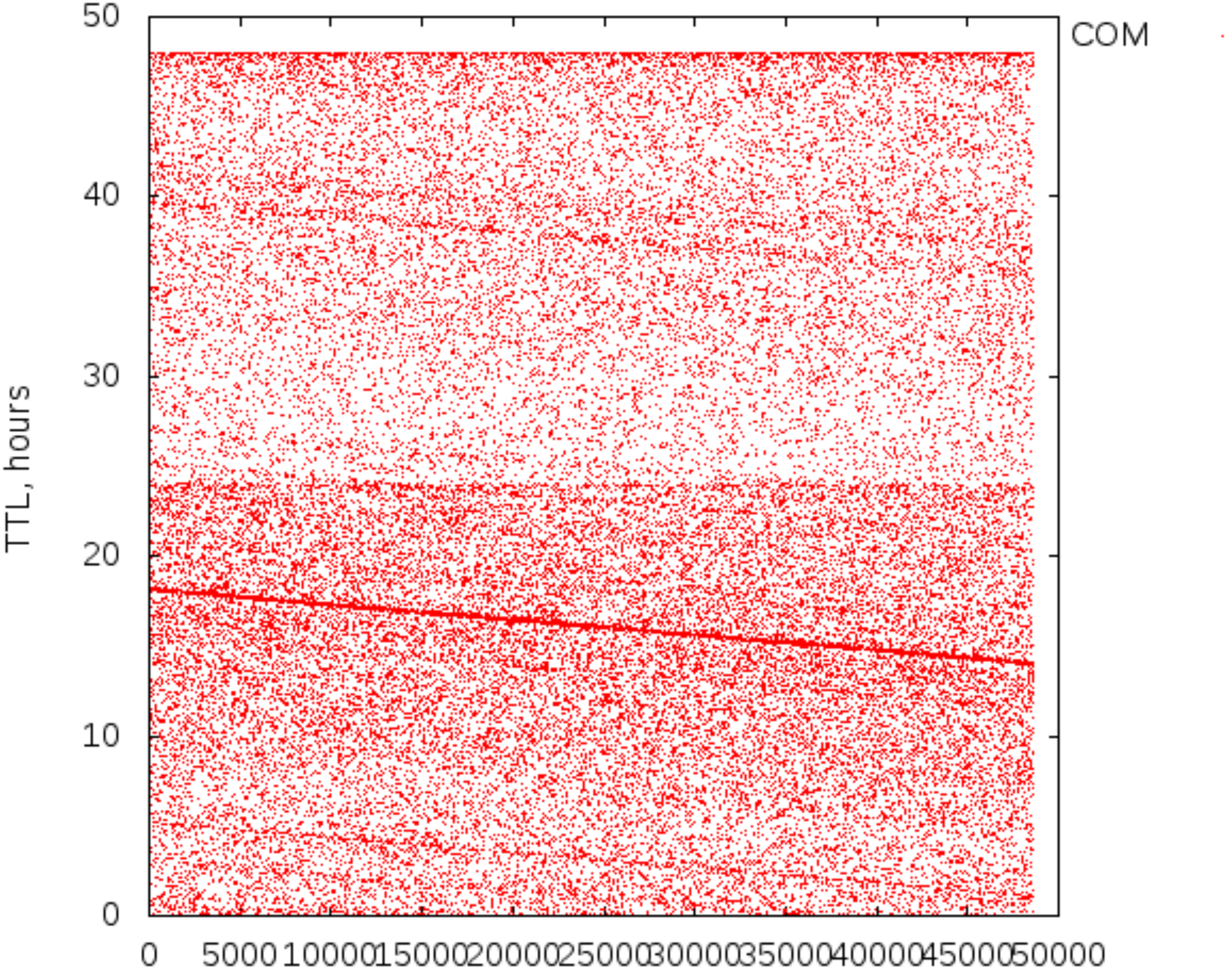
Are TTLs Uniformly Distributed in the Internet's DNS Caches?

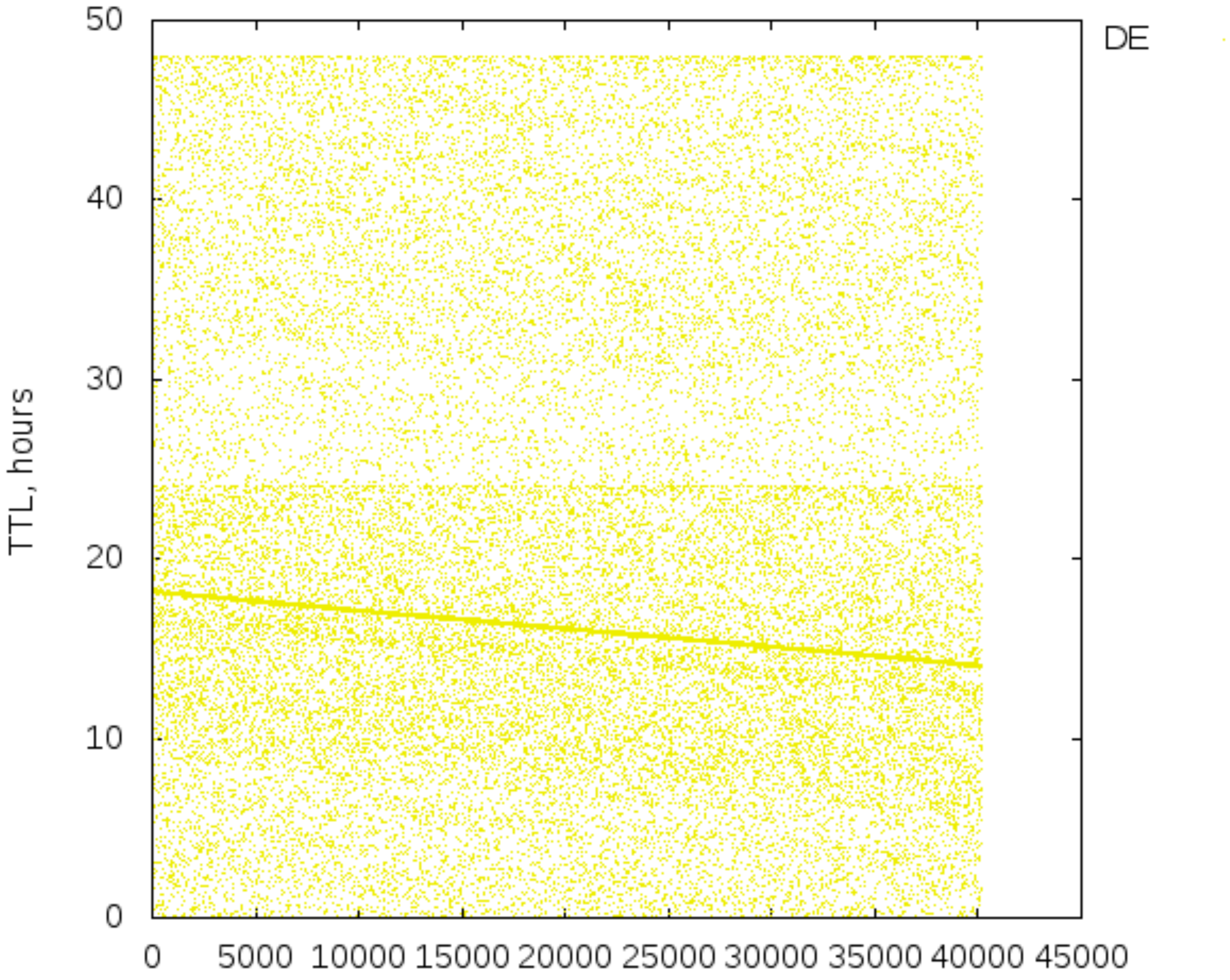
Let’s Ask Some Open Resolvers

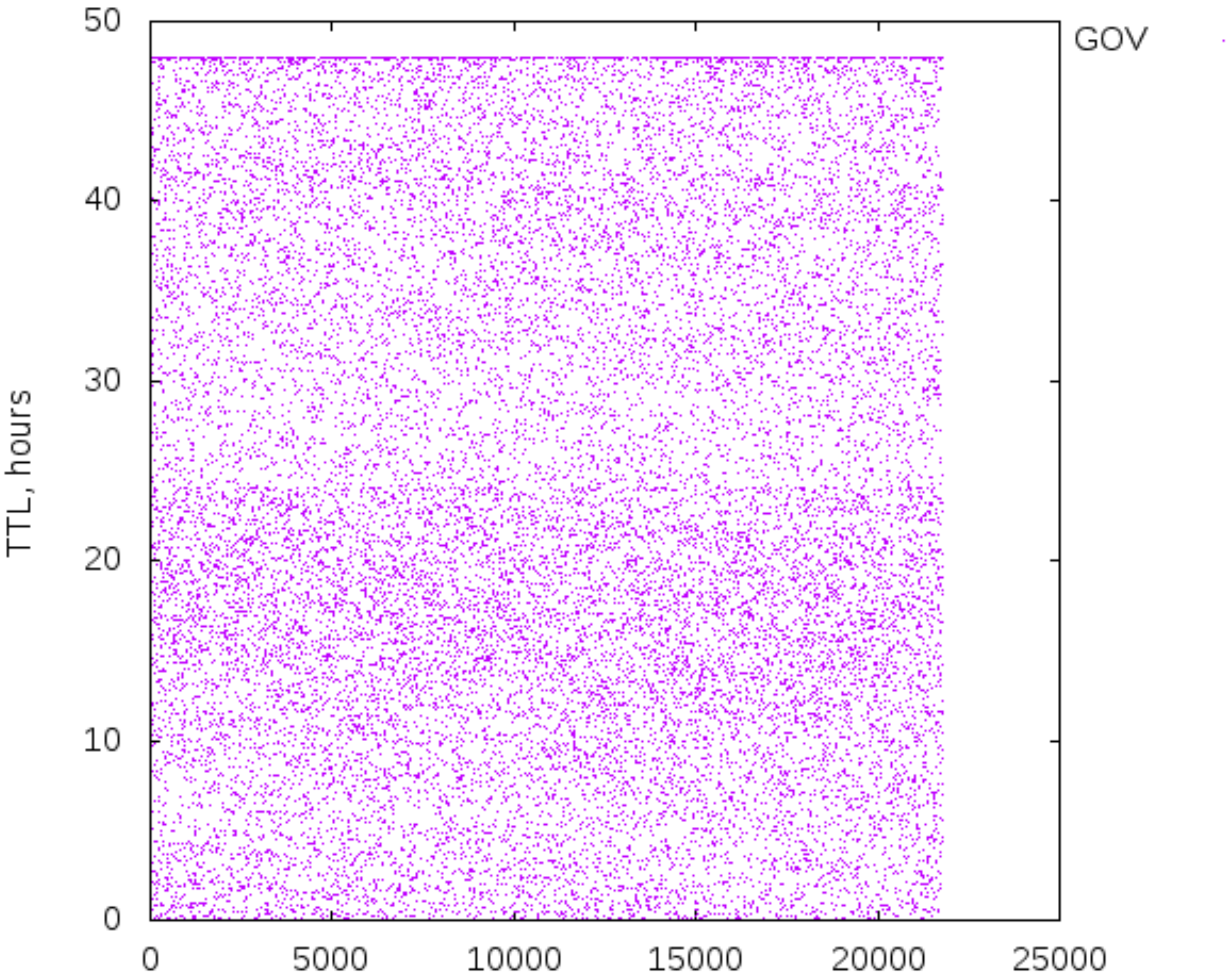
- Sent queries to 53,000 open resolvers
- Asked for NS records of com, net, org, gov, uk, de, cn, au
- Recorded minimum TTL for each response NS RRset

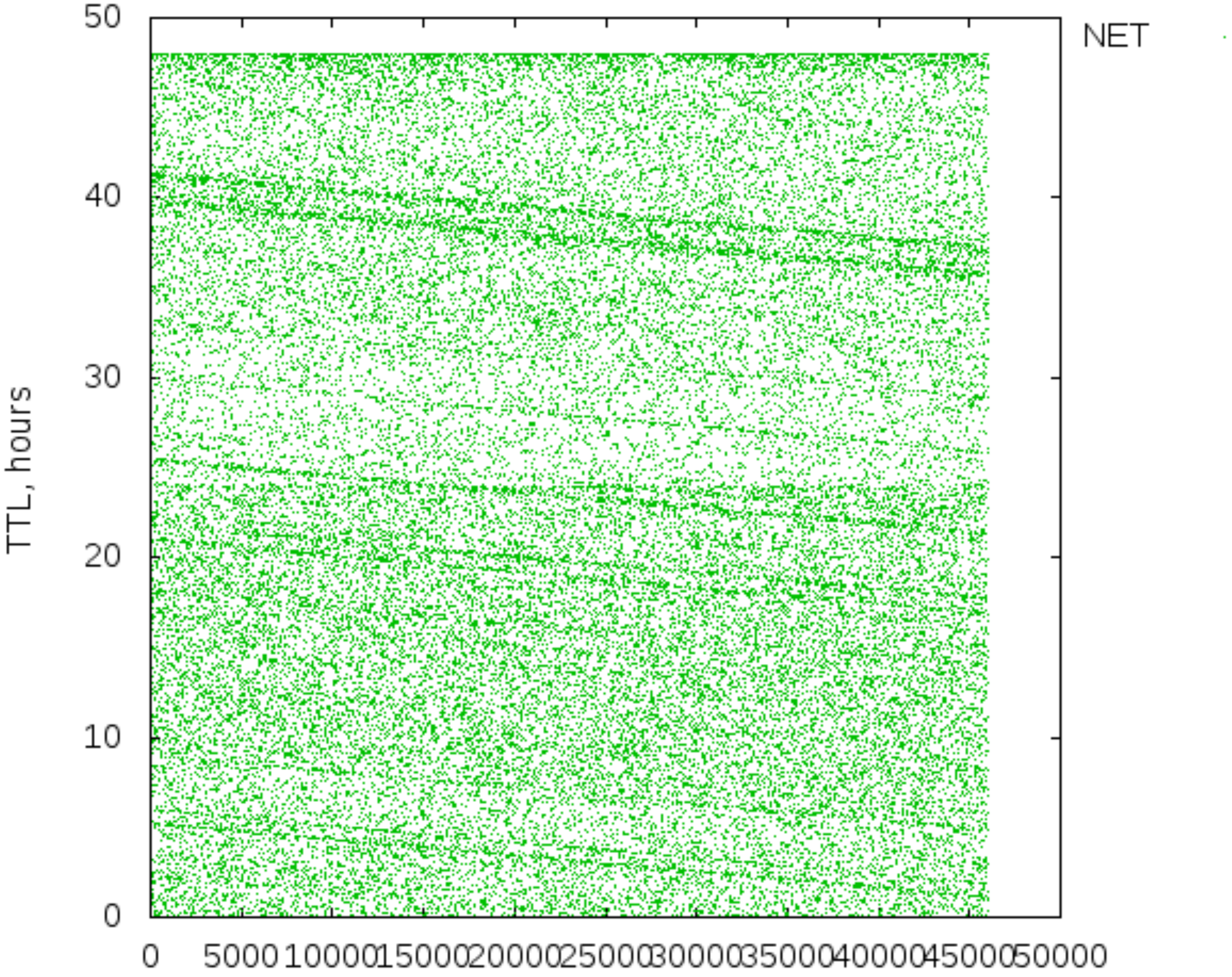

50
40
30
20
10
0
TTL, hours
0
5000
10000
15000
20000
25000
30000
AU

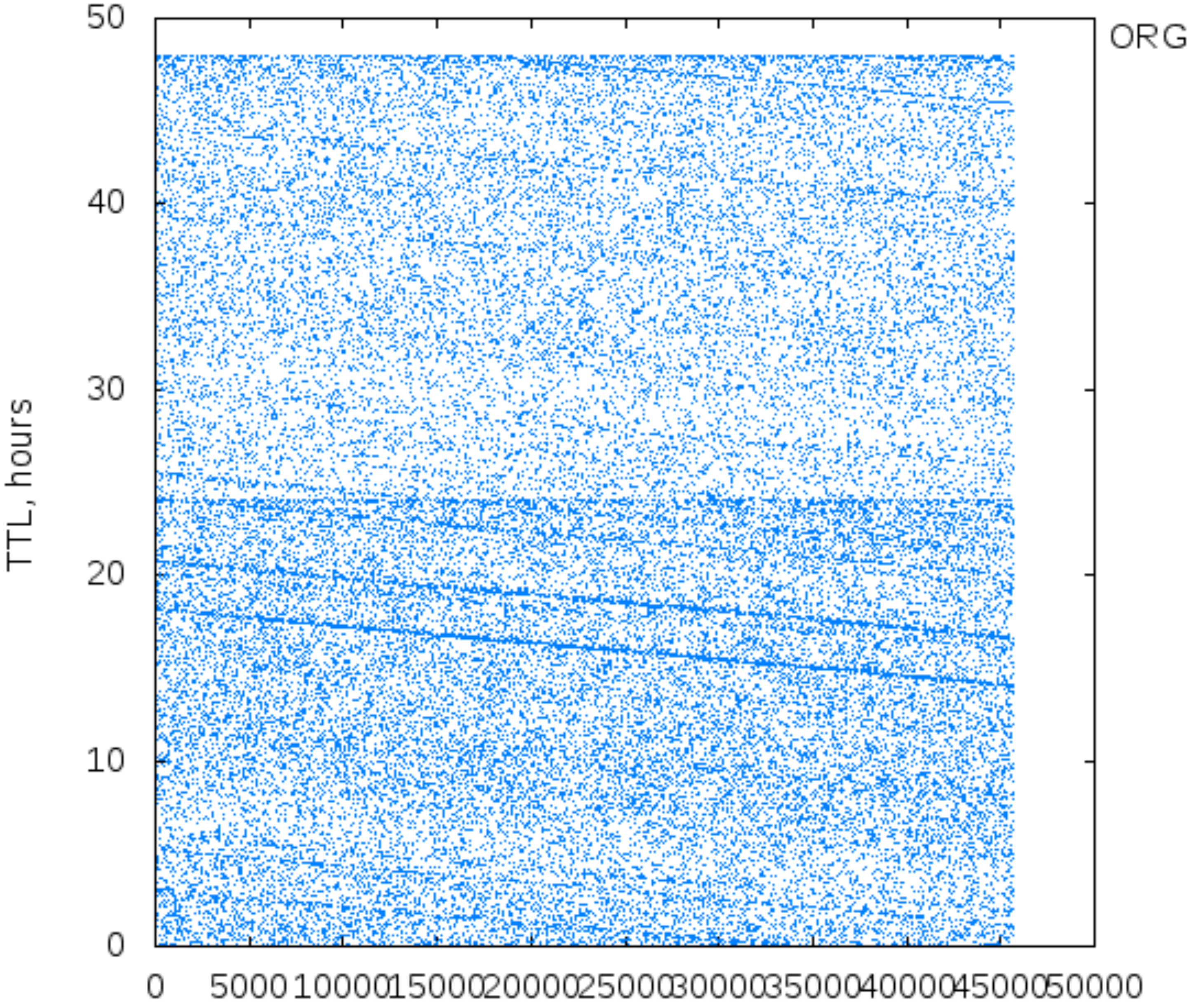

TTL, hours
0
10
20
30
40
50
0
5000
10000
15000
20000
25000
30000
35000
CN

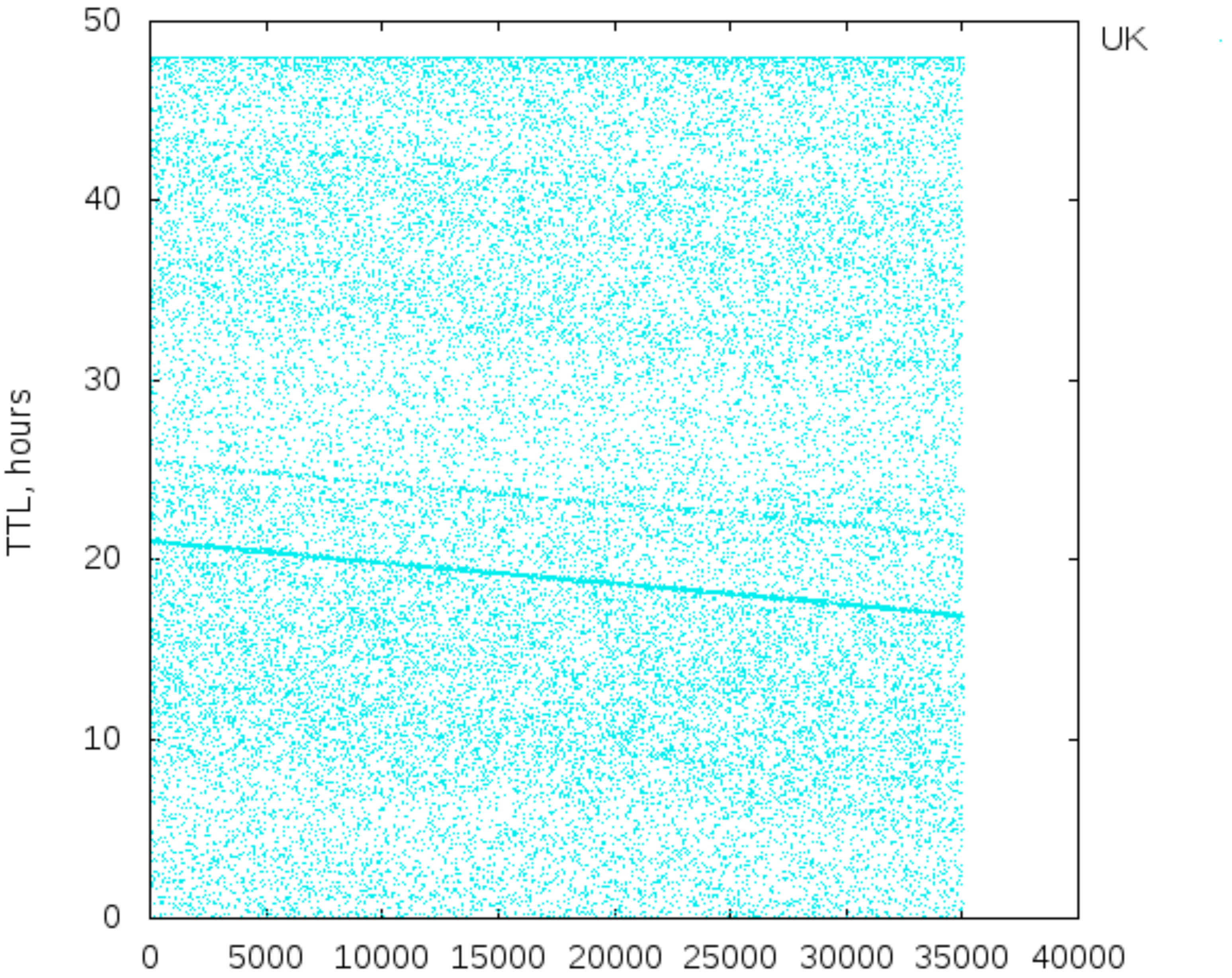

TTL, hours
COM
0
10
20
30
40
50
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


TTL, hours
0
10
20
30
40
50
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
DE

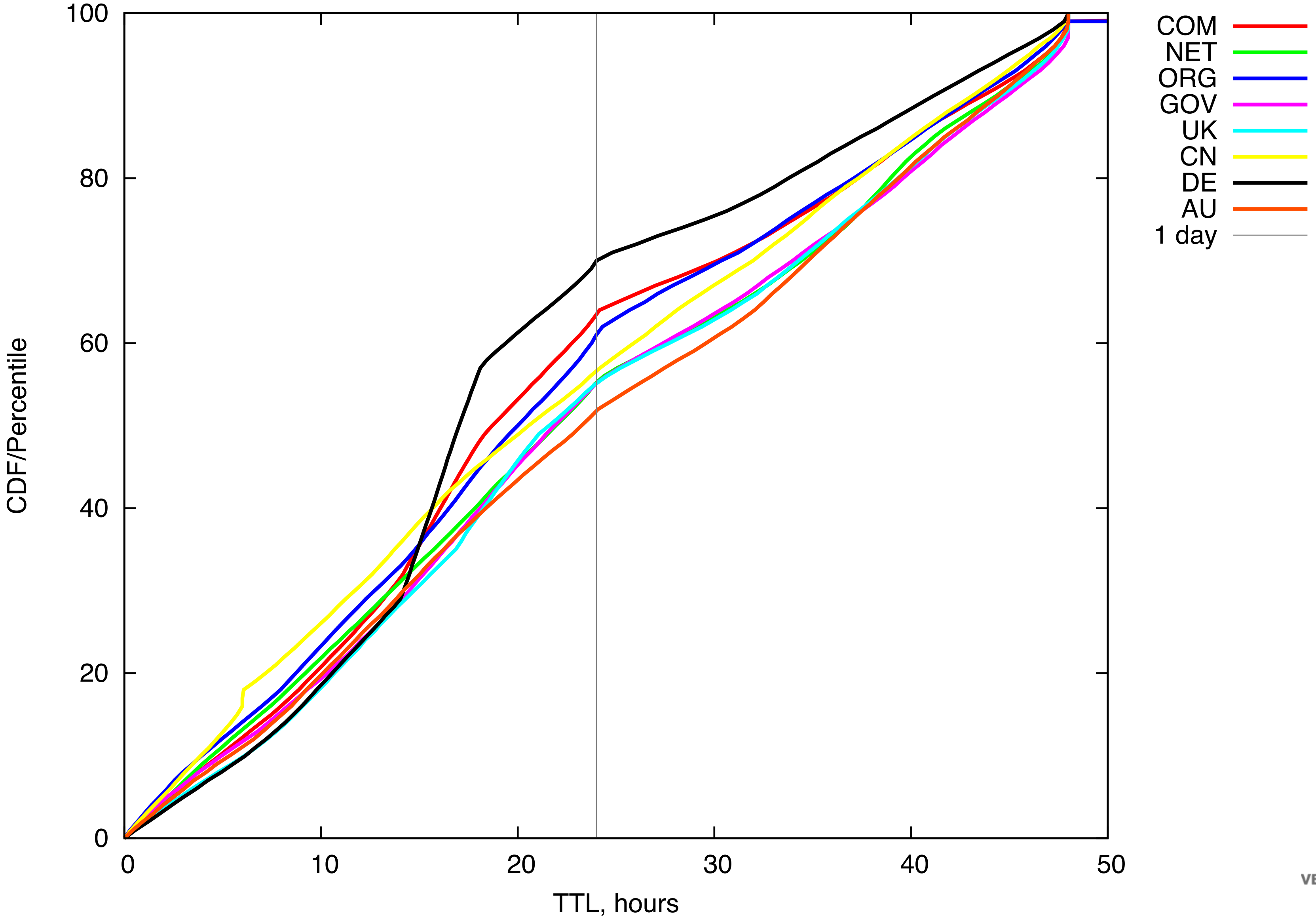

50
40
30
20
10
0
TTL, hours
0
5000
10000
15000
20000
25000
GOV


NET
TTL, hours
0
10
20
30
40
50
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


ORG
TTL, hours
0
10
20
30
40
50
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


UK
TTL, hours
0
10
20
30
40
50
0
5000
10000
15000
20000
25000
30000
35000
40000

Cumulative Distributions



What TTLs are Actually In Use by TLDs?

VERISIGN™

COM

Type	TTL
NS	2 days
A	2 days
AAAA	2 days
DNSKEY	1 day
RRSIG	2 days

ORG

Type	TTL
NS	1 day
A	1 day
AAAA	1 day
DNSKEY	15 min
RRSIG	1 day

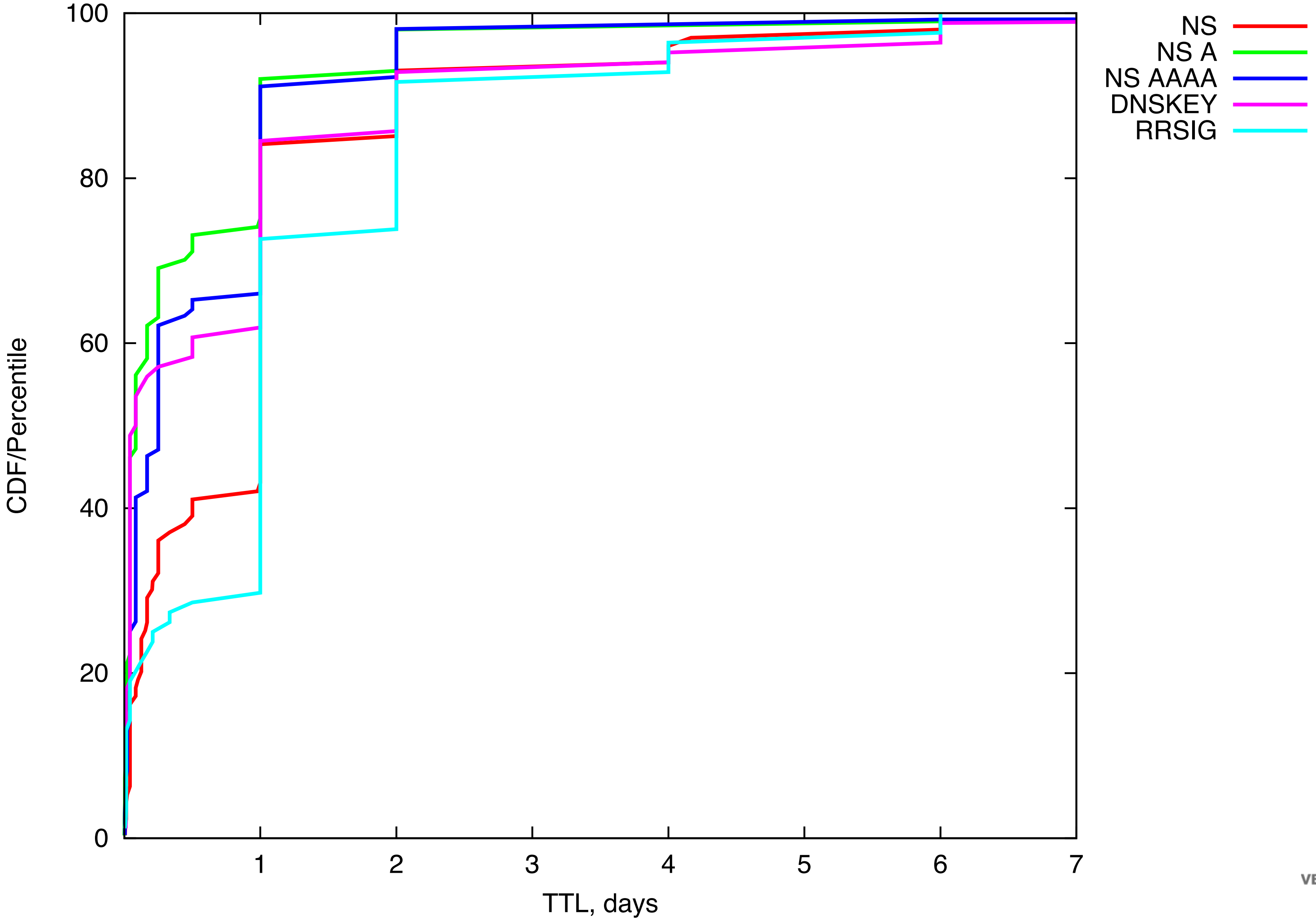
MUSEUM

Type	TTL
NS	1 hour
A	2 hours
AAAA	2 hours
DNSKEY	6 hours
RRSIG	1 hour

ES

Type	TTL
NS	2 hours
A	1 hour
AAAA	1 hour
DNSKEY	-
RRSIG	-

TLD TTL CDF


100
80
60
40
20
0
CDF/Percentile
1
2
3
4
5
6
7
TTL, days
NS
NS A
NS AAAA
DNSKEY
RRSIG

I’ve Heard Some Resolvers Have Upper Limits on TTLs

VERISIGN™

Resolver TTL Maximums

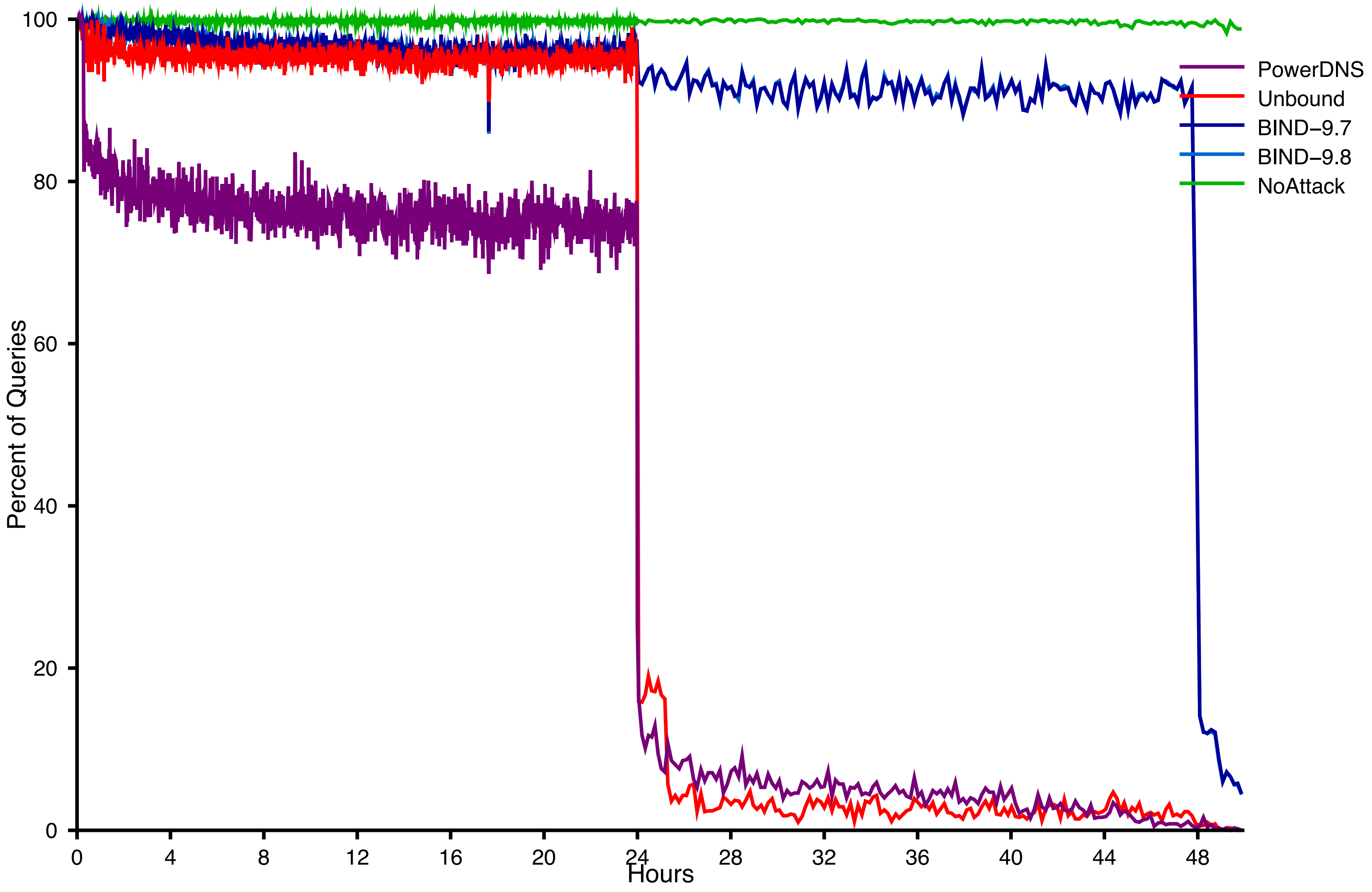
Implementation	Directive	Default
BIND	max-cache-ttl	7 days
Unbound	cache-max-ttl	1 day
PowerDNS Recursor	max-cache-ttl	1 day

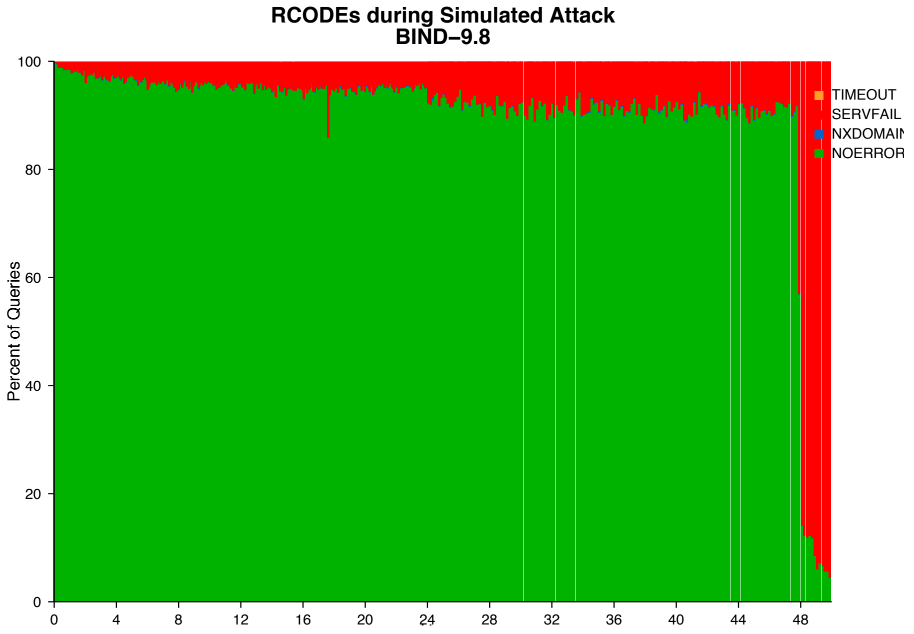
Does It Really Work?

Simulating A Root Zone Outage

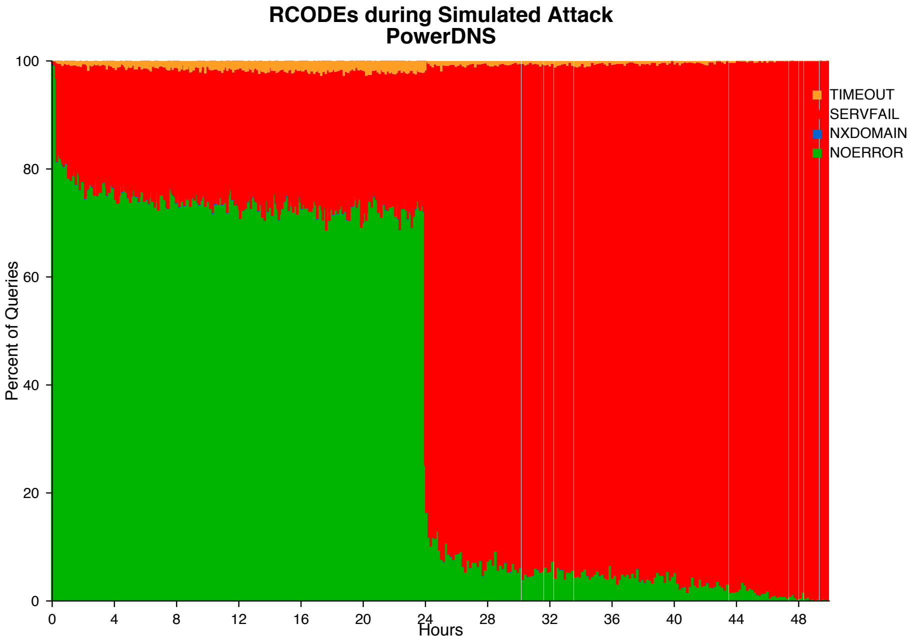
- Run BIND (9.8.0-P2), Unbound (1.4.16), PowerDNS (3.3) on different machines.
  - plus a fourth (running BIND) as the “No Attack” control.
- Take a trace of (resolvable) query names (and types) from com/net name servers.
  - Query names end in either .com or .net. Their NS names may be in other TLDs, however.
- Prime resolver caches with SOA queries for every TLD.
- Replay trace, sending each query to all 4 resolvers at the same time.
- Block queries to Root Server IP addresses (static route to loopback) 10 minutes into the trace.

Percent of Queries Answered Successfully

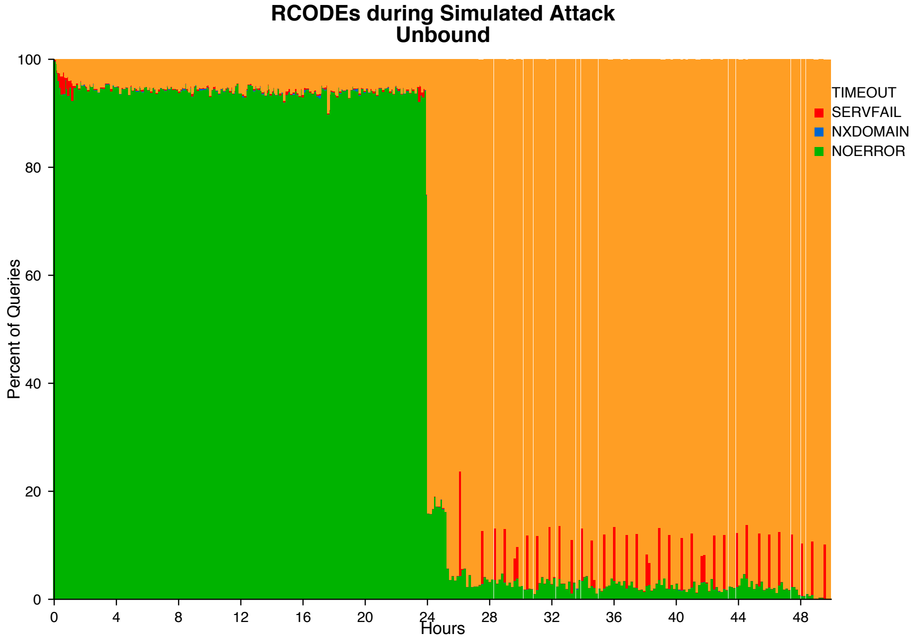




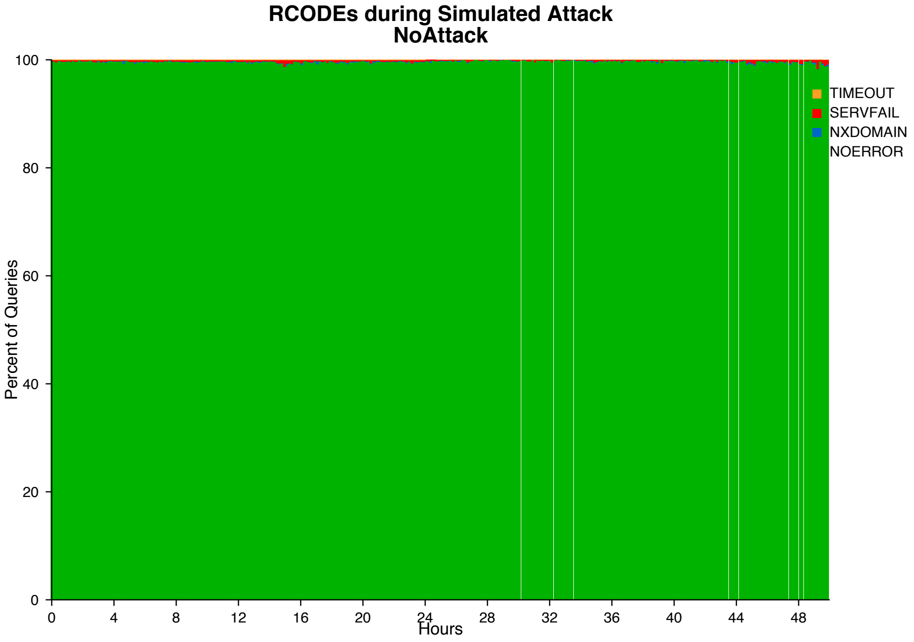
RCODEs during Simulated Attack
BIND−9.8



RCODEs during Simulated Attack
PowerDNS



RCODEs during Simulated Attack
Unbound



RCODEs during Simulated Attack
NoAttack

Thank You

VERISIGN™