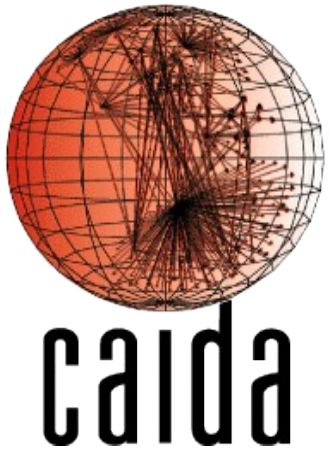
DITL 2009

Analysis and results of four years of DITL

Sebastian Castro
sebastian@nzrs.net.nz

NZRS / CAIDA



p: (+64) (4) 931 6977 e: registry@nzrs.net.nz
f: (+64) (4) 931 6979 w: www.nzrs.net.nz
P.O. Box 24361, Wellington, New Zealand
New Zealand Domain Name Registry Limited

Overview

- Participants
- General statistics and trends
- EDNS from different angles
- SPR evolution
- Conclusions

Participants

- More successful than 2008!
  - 8 Root servers: A, C, E, F, H, K, L, M
  - 7 TLDs: BR, CL, CZ, INFO, NO, SE, UK
  - 3 RIRs: APNIC, ARIN, LACNIC
  - 5 instances of AS112 servers
  - Packet Pushers
  - SWITCH
- Longer than previous years
  - From March 30 to April 1
  - For analysis purposes, we selected the best 24 hour

General Statistics table

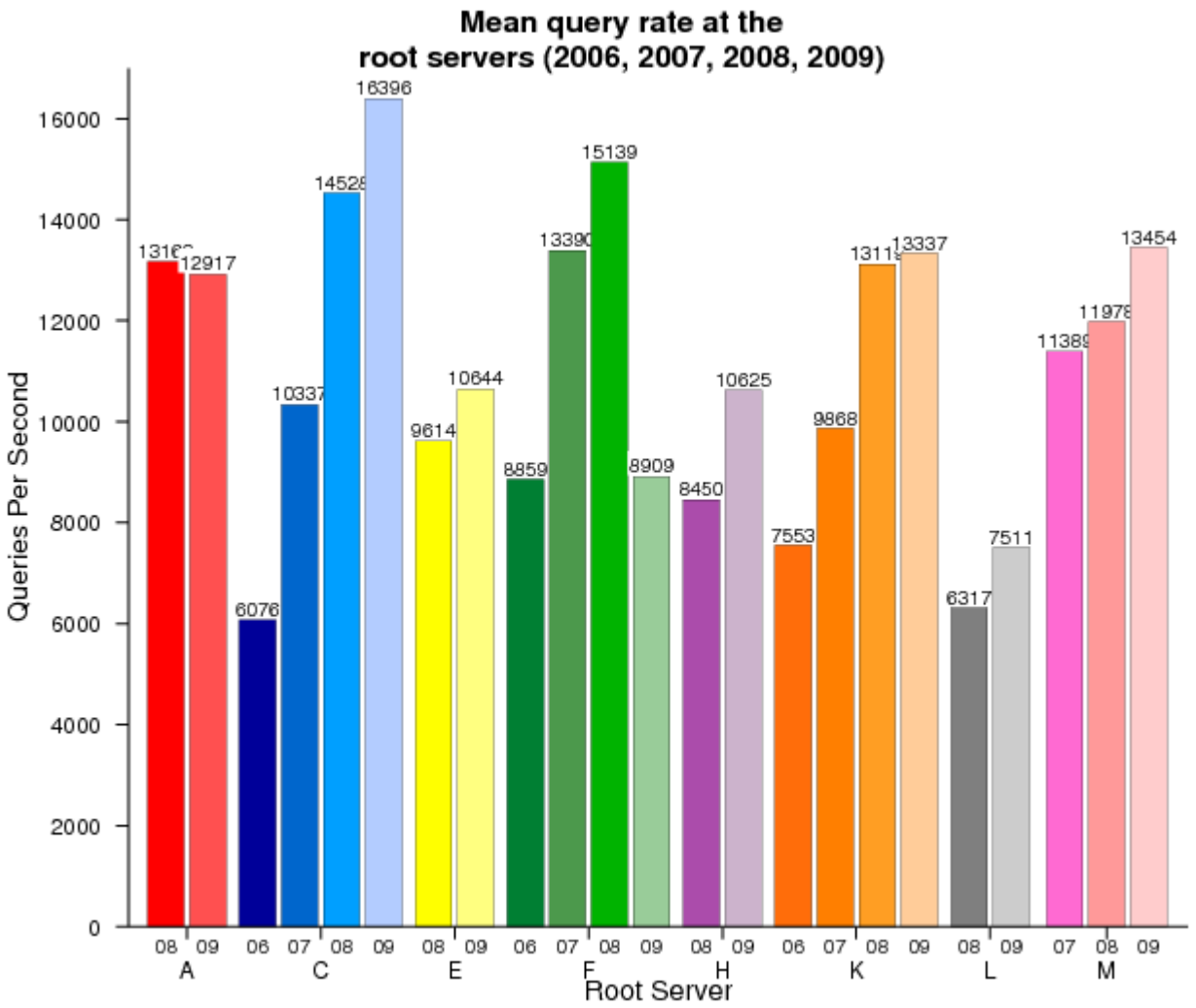
	DITL 2007	DITL 2008	DITL 2009
Dataset duration	24h	24h	24h
Dataset start (UTC)	January 9, noon	March 19, midnight	March 31, midnight
Number of instances	C: 4/4 F: 36/40 K: 15/17 M: 6/6	A: 1/1 C: 4/4 E: 1/1 F: 35/41 H: 1/1 K: 15/17 L: 2/2 M: 6/6	A: 1/1 C: 6/6 E: 1/1 F: 36/48 H: 1/1 K: 16/17 L: 2/2 M: 6/6
Query count	3.84 billion	7.99 billion	8.09 billion
Unique clients	~2.8 million	~5.6 million	~5.8 million
Recursive queries	17.04%	11.99%	9.76%

General Statistics table

	DITL 2007	DITL 2008	DITL 2009
TCP Bytes (1)	1.65%	0.86%	0.74%
TCP Packets (1)	2.67%	1.45%	1.20%
TCP Queries (1)	~700K	~2.07 million	~3.04 million
Queries over IPv6	~228K	~23 million	~29 million
Number of instances providing IPv6 traffic	6	16	31
Queries from RFC 1918 (2)	4.26%	1.31%	1.57%
Queries from bogon address space (3)	0.05%	0.35%	0.04%

(1) L-root did not collect TCP traffic
(2) A, E, K and L-root did not see any traffic (filter in place?)
(3) A-root did not see any traffic (filter in place?)

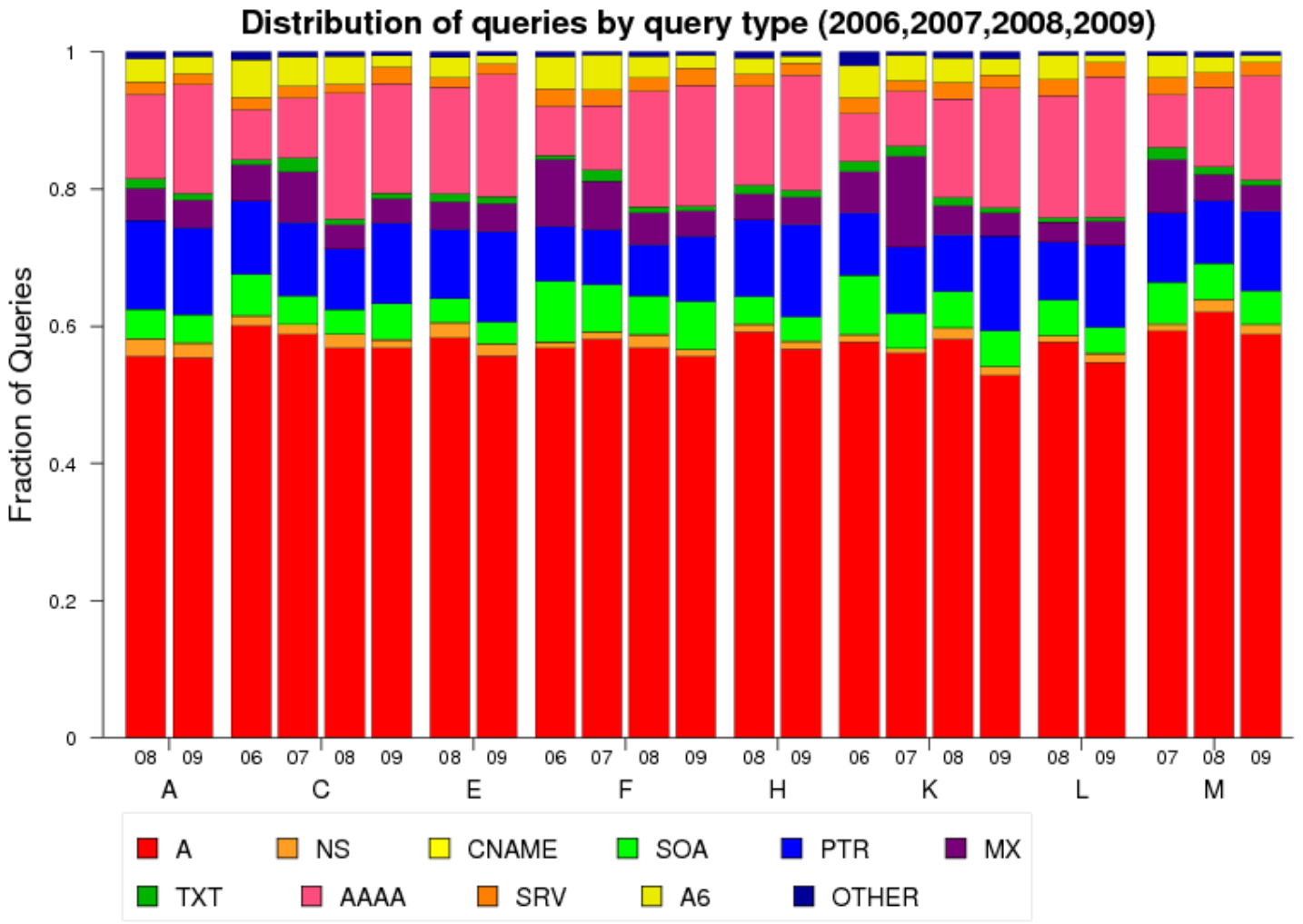
Mean query rate growth



Root	Growth 2007-2008	Growth 2008-2009
A		-1.91%
C	40.54%	12.86%
E		10.71%
F	13.06%	-41.15% *
H		25.74%
K	32.94%	1.66%
L		18.90%
M	5.17%	12.32%

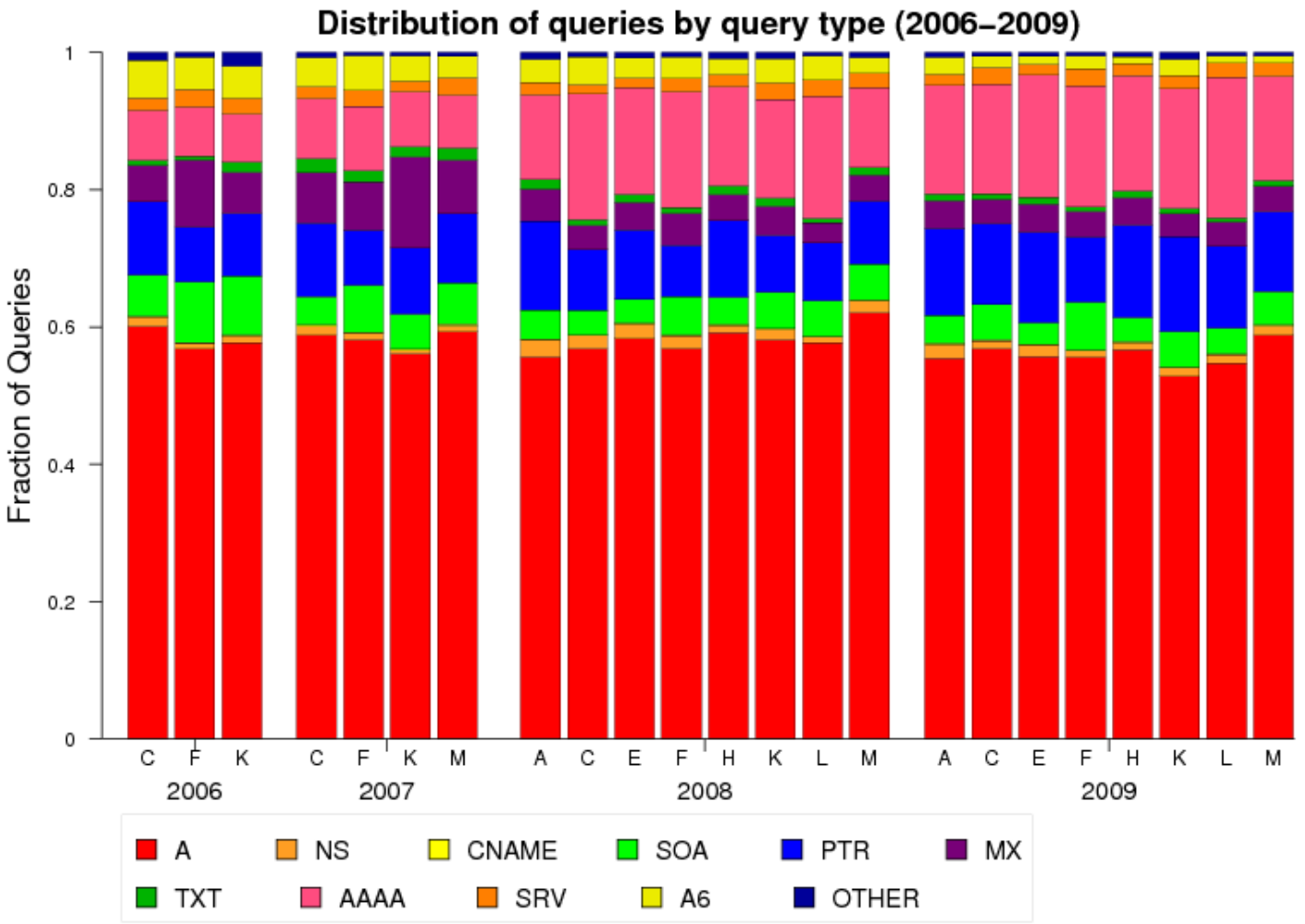
* Data from f-sfo (global instance) was not collected

Query type distribution



- A-queries are still the most popular
- AAAA-queries has gained popularity

Query type distribution

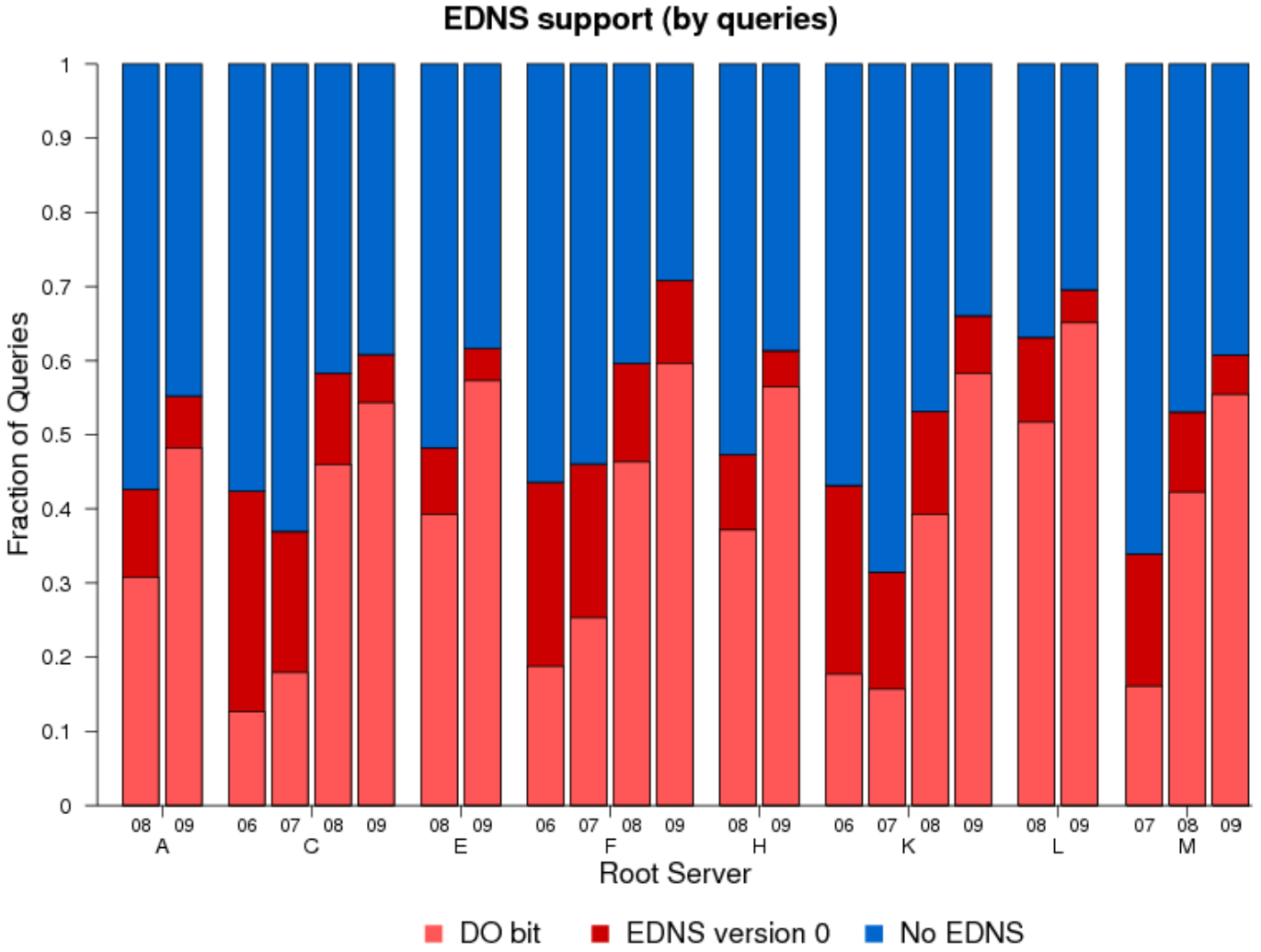


- This graph shows the distribution by query type is similar across the roots
  - And the evolution across the years for the different query types

EDNS (outline)

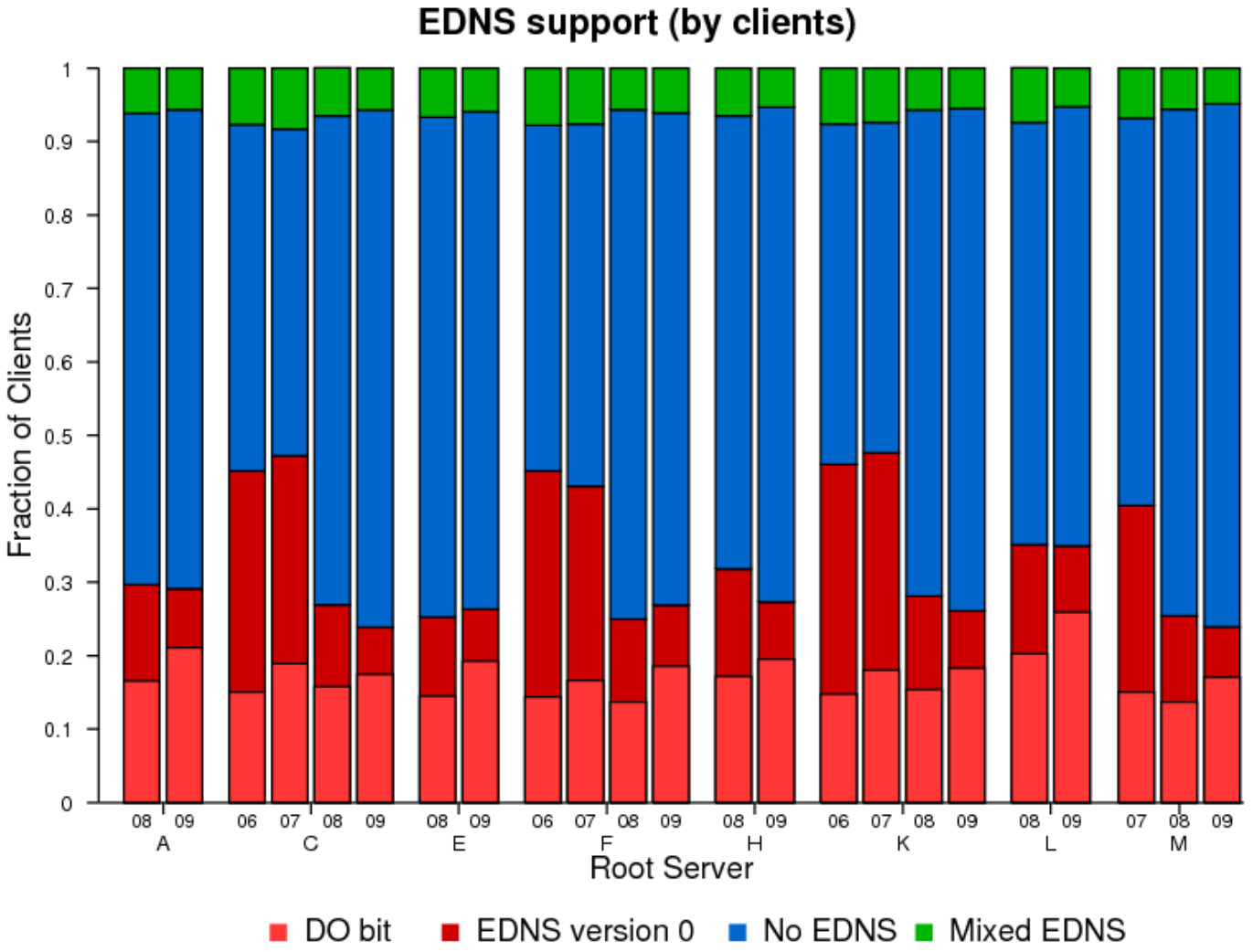
- We provide two metrics to estimate the presence of EDNS support on the traces
  - At the query level
    - By checking the presence of the OPT RR and analyze their values.
    - Can be: No EDNS, EDNS0 (with or without DO bit), EDNSX
  - At the client level (check all the queries for a source address and tag them accordingly)
    - Tags: No EDNS, EDNS0, mixed (not all queries include EDNS support)

EDNS at the query level



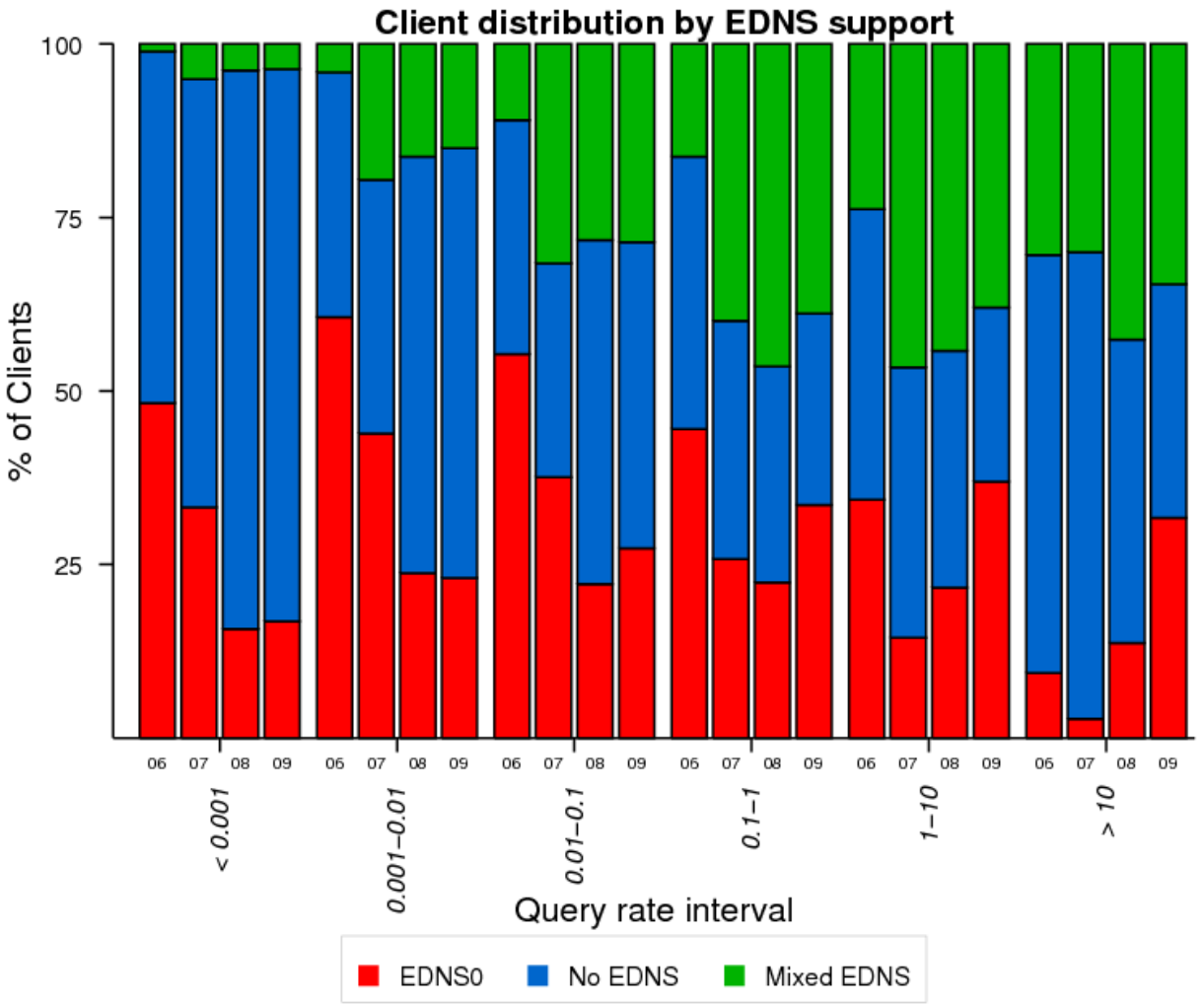
- Clear growth of EDNS, with a jump from '07 to '08
- Over 90% of the EDNS capable queries are DO enabled in 2009
- Good news, right?

EDNS at the client level



- At the client level, the situation is totally the opposite!
  - Reduced along the years
  - Around 30% support
  - The DO enabled/EDNS capable queries ratio is in the 60-70% range
  - How is this explained?
    - **The heavy hitters**

EDNS per query rate



- How we can explain the difference?
  - We grouped the clients by their query rate
  - Clients sending few queries present less EDNS support
    - And they represent most of the clients
  - Client sending lots of queries present more EDNS support
    - Most of the queries (>50%) are generated by the two rightmost categories
    - Most of their queries are pollution :(

EDNS mixed support

- The percentage of 'green' clients is in the range of 6-8% the total
  - For 2009, we have 396,046 unique sources with mixed EDNS (7%)
- There are two reasons why a client can show mixed support
  - Several hosts behind the same address (NAT?)
  - A client fails to receive response to queries with EDNS support, so retries without it
- We checked for “EDNS fallback” patterns
  - Sequences of the same query <QNAME, QCLASS, QTYPE>
  - Separated by a short period of time (<10 sec)
  - Retrying with EDNS enabled until the last query has EDNS disabled
- Hard because
  - Missing data: a client retrying on a root server we don't have data for
  - Elapsed time between attempts is not regular

EDNS fallback detection

How a EDNS fallback sequence looks like?

- The same query, at regular intervals
- Attempts with the OPT RR (1au)
- Ends with the same query but without the OPT RR. The query is not seen again for 10 seconds or more

00:00:51.960047 IP 200.91.16.75.32768 > 6.0.0.25.53: 3092% **[1au]** A? NS4.NIC.CO. (39)

00:00:54.023484 IP 200.91.16.75.32768 > 3.0.0.2.53: 517% **[1au**] A? NS4.NIC.CO. (39)

00:00:56.061242 IP 200.91.16.75.32768 > 13.0.0.1.53: 642% **[1au]** A? NS4.NIC.CO. (39)

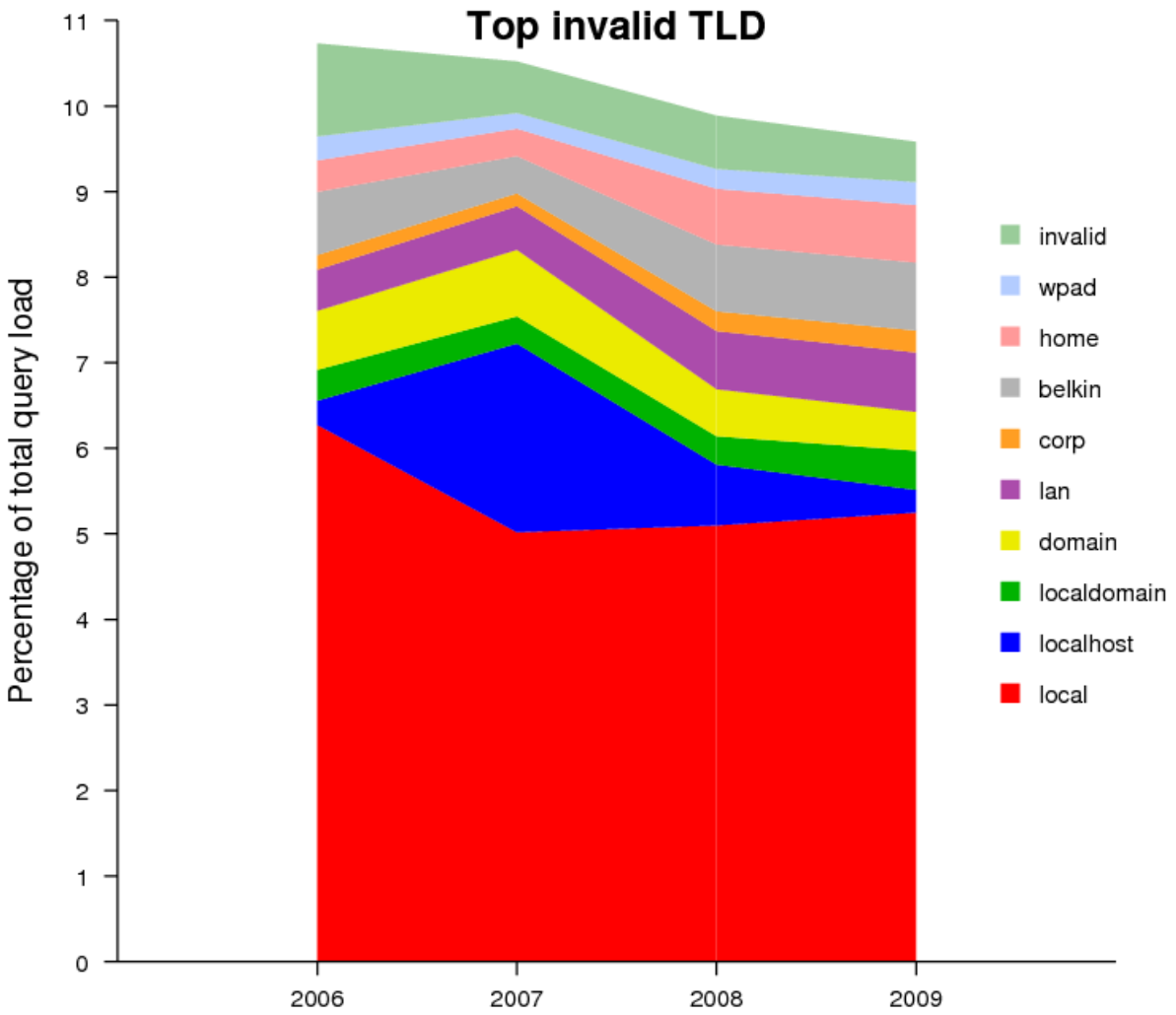
00:00:58.057185 IP 200.91.16.75.32768 > 1.0.0.1.53: 31143 A? NS4.NIC.CO. (28)

- Tested the queries for 72,000 unique sources looking for this pattern using the traces from the roots collected in DITL 2009
  - Our sample represents 18% of the total mixed clients
  - Each coincidence counts as a sequence
  - 25,773 source addresses (35.8% of the sample) presented at least one sequence

Root priming and EDNS

- We also explored the queries that looked like “root priming”
  - QNAME=”.”, QTYPE=NS
- Verifying for EDNS fallback sequences
- 1,3 million unique sources sent this type of queries
  - ~900K not EDNS capable
  - ~355K EDNS capable
  - ~40,000 with mixed EDNS
    - ~10,000 clients matched the detection sequence

Traffic for invalid TLDs

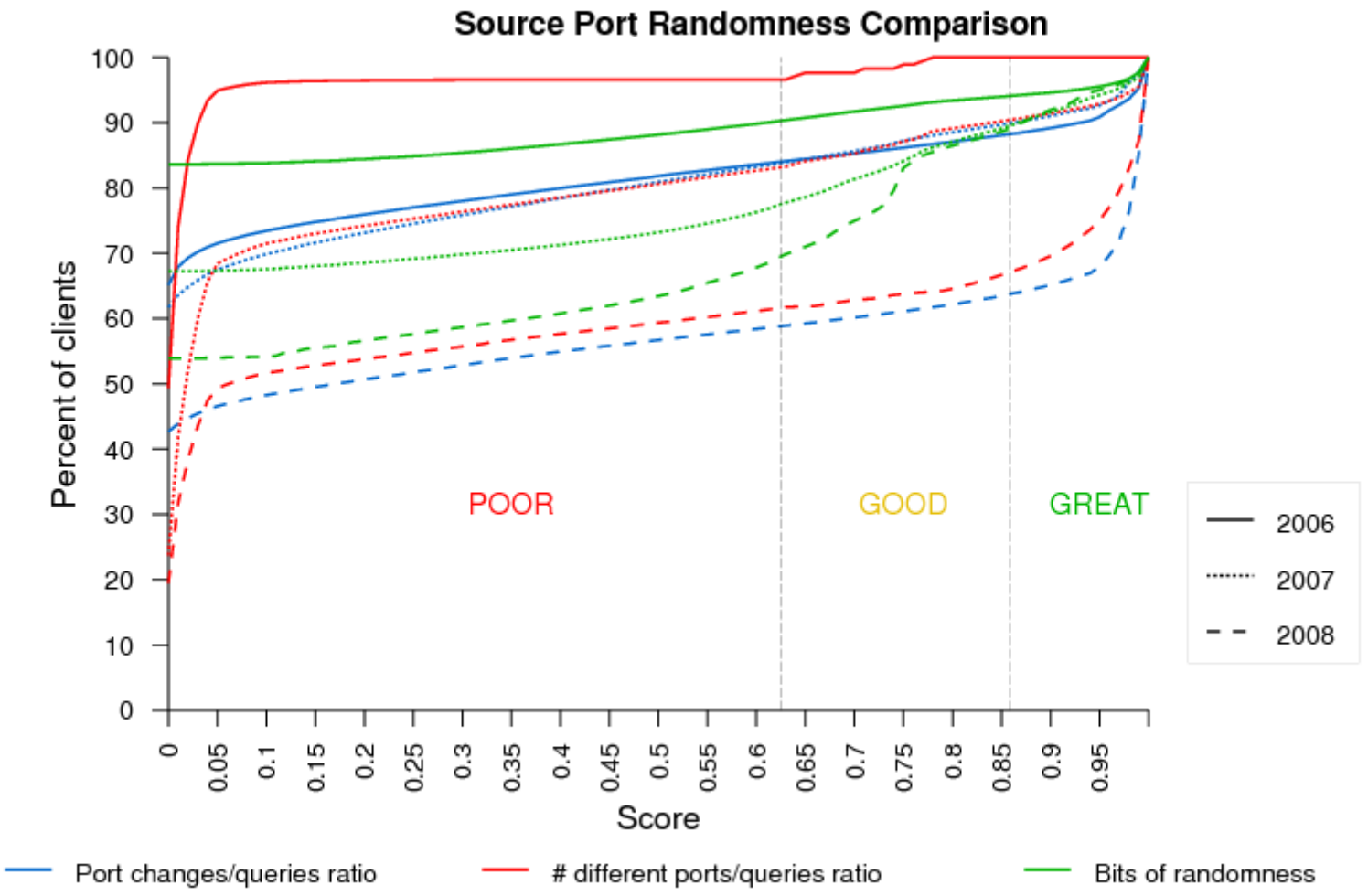


- 10 invalid TLDs represent 10% of the **total** query load at the root servers
- The TLD has not changed in the last four years (only the ranking)
- If all invalid TLDs are included, the percentage moves from 18% to 26% (not shown)

Source Port Randomization evolution

- Not new, but useful to observe trends
- Three scores for each client (if # queries > 20)
  - # of port changes/queries ratio
  - # different ports/queries ratio
    - These two proposed by nic.at
  - Bits of randomness
    - Proposed by Duane Wessels
- Based on the scores, each client is tagged
  - If score < 0.62, tagged as “Poor”
  - If score in [0.62, 0.86], considered “Good”
  - If score > 0.86, considered “Great”

SPR (cont)


Source Port Randomness Comparison
Percent of clients
100
90
80
70
60
50
40
30
20
10
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
Score
POOR
GOOD
GREAT
2006
2007
2008
Port changes/queries ratio
different ports/queries ratio
Bits of randomness

Conclusions

- The DITL data collection is a very useful source of data to better understand the authoritative side of DNS at different levels
  - Our analysis are focused on the roots, with little exploration of the TLDs data
- Between 25-35% percent of the clients with mixed EDNS are showing EDNS fallback behavior
  - Although represents less than 1% of the clients at the roots, the numbers are based on a sample
- Source Port Randomization has improved along the years
  - Is it yet good enough?
-

Thoughts

- The DITL collection and data is likely to be the right place to look for answer to deployment questions
  - DNSSEC for example?
- More brain and computing power is needed to extract particular information
  - Contact OARC if you are interested on access to the data
- May be we should share the analysis code?
  - We have solved some issues (or found workarounds) to handle large datasets