On Measuring the Client-Side DNS Infrastructure

Kyle Schomp†, Tom Callahan†, Michael Rabinovich†, Mark Allman†‡

†Case Western Reserve University

‡International Computer Science Institute

Motivation

- DNS provides the mapping between human friendly names and machine friendly addresses
  - amazon.com -> 1.2.3.4
- DNS resolution path is both complex and hidden
  - Multiple layers of resolvers
  - Controlled by different organizations
  - No clear attribution if something goes wrong

Our Contribution

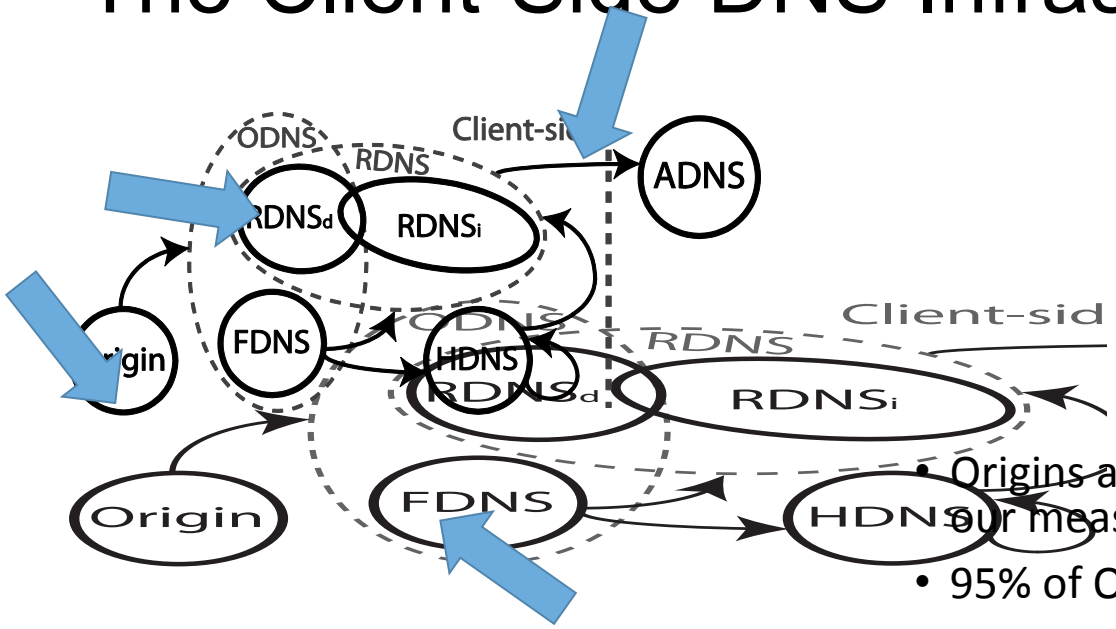
- Methodologies for discovering the client-side DNS infrastructure
- Measurement techniques for teasing apart behavior of various actors
- Application of our methodologies and techniques to assess behavior
  - How long are records retained in caches
  - How time-to-live (TTL) values a modified by resolvers

We have also used our methodologies to study security properties of DNS. This is a separate work that is not discussed today.

Discovery Methodology

- We randomly sample IP addresses from the Internet
- To each sampled IP address, we send DNS requests looking for open resolvers
- We also deploy an authoritative DNS server
- Our DNS request probes target our own domain
- We can collect both the ingress and egress servers of the client-side DNS infrastructure

The Client-Side DNS Infrastructure



Structure of the client-side DNS infrastructure observed in our datasets.

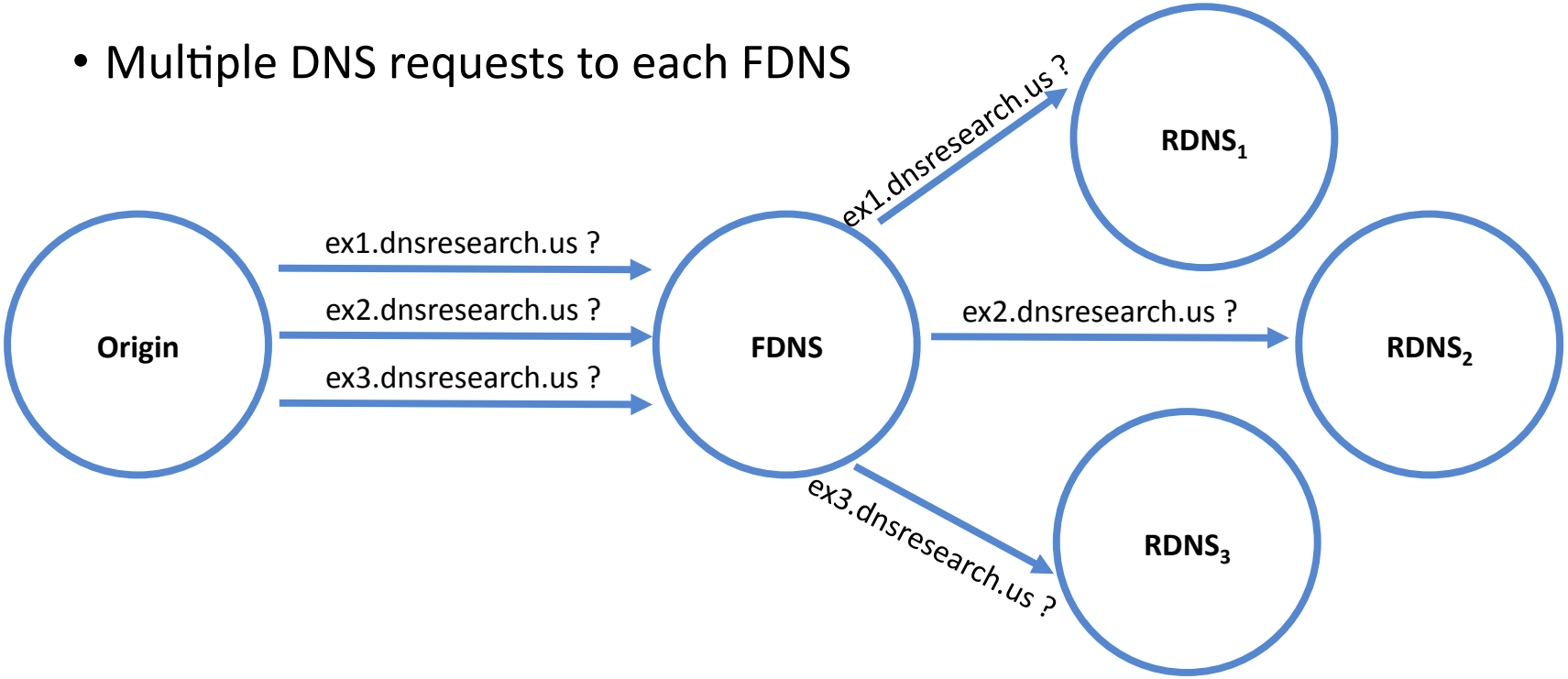
- Origins are either end user devices or our measurement points
- 95% of ODNS are FDNS
- 78% of ODNS are likely residential network devices

RDNS Discovery

- 2/3 of RDNS in our datasets are closed
  - Do not respond to direct probes
  - Must be discovered through FDNS
- Two techniques for RDNS discovery
  - Multiple DNS requests to each FDNS
  - CNAME “chains” from our ADNS

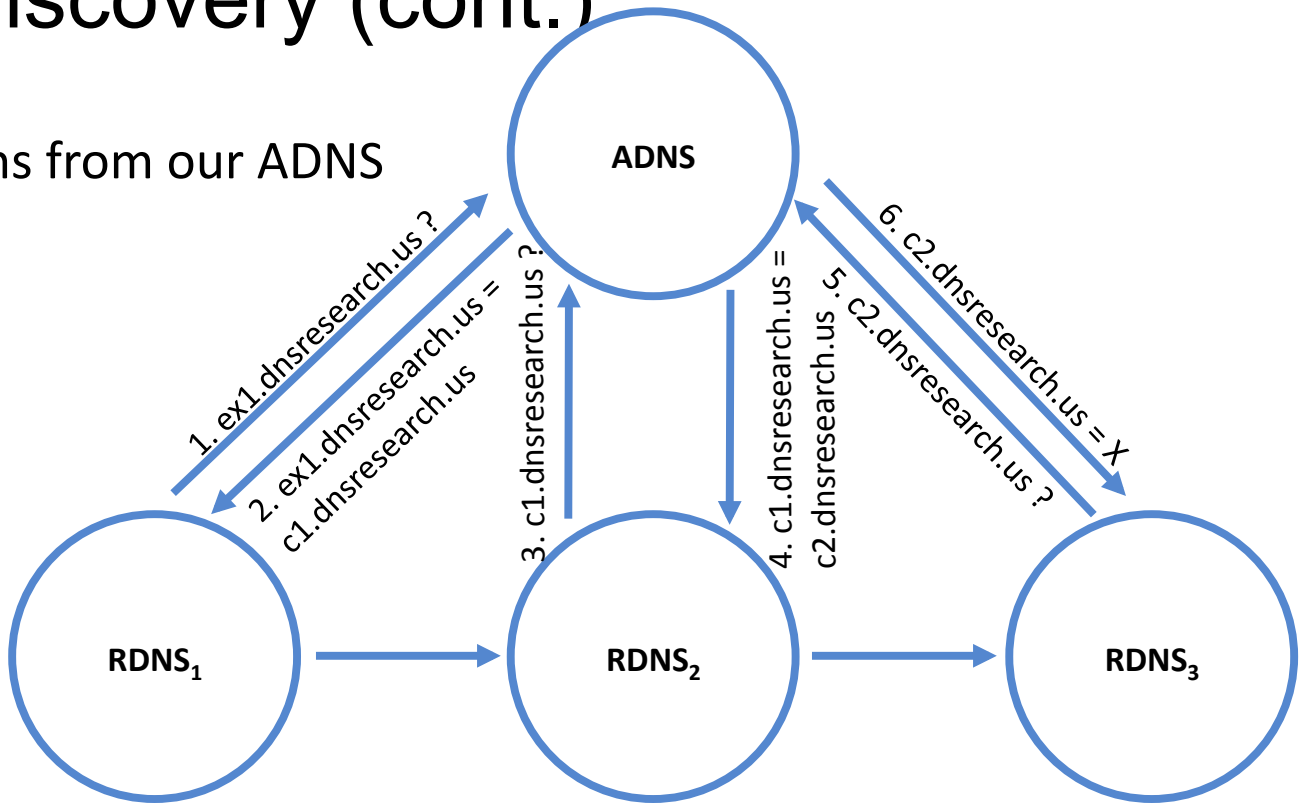
RDNS Discovery (cont.)

- Multiple DNS requests to each FDNS



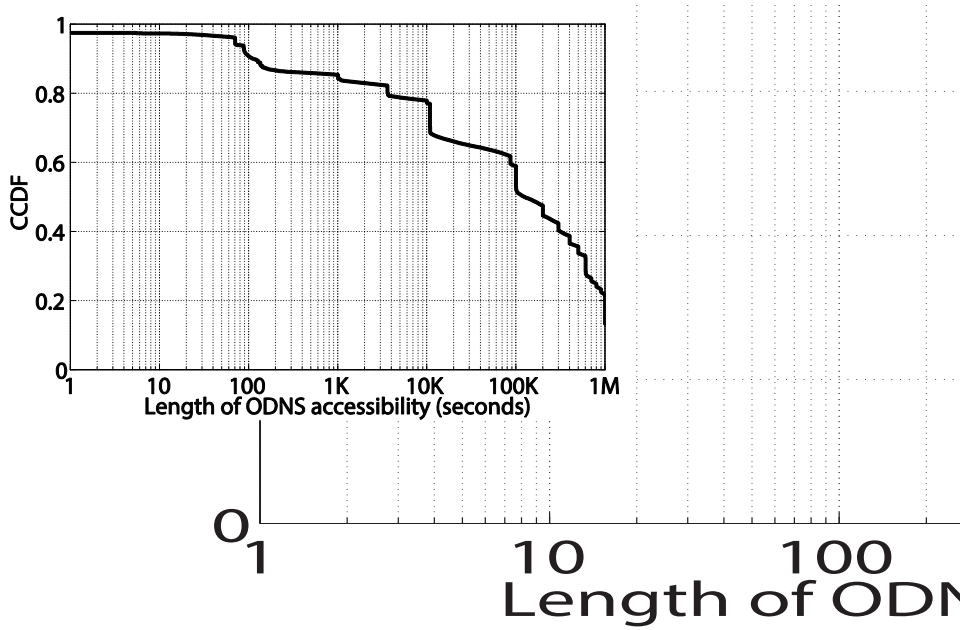
RDNS Discovery (cont.)

- CNAME chains from our ADNS



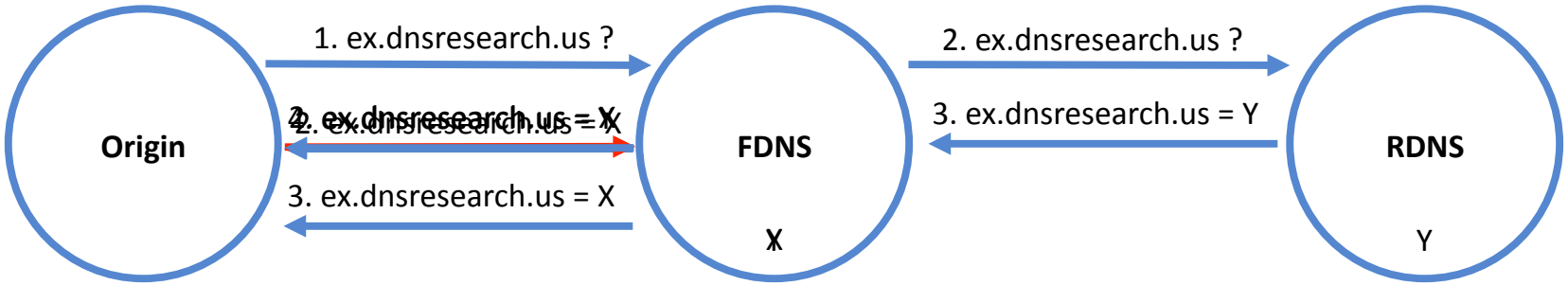
Measurement Principles

- Non-Interference with Normal Operation
  - Probe for our own domain only
  - Limit probing rate
- ODNS Short Lifetime
  - Experiment during discovery
- Random bindings
  - Two requests for the same domain will receive different bindings with high probability



Measuring FDNS (Cache Injection)

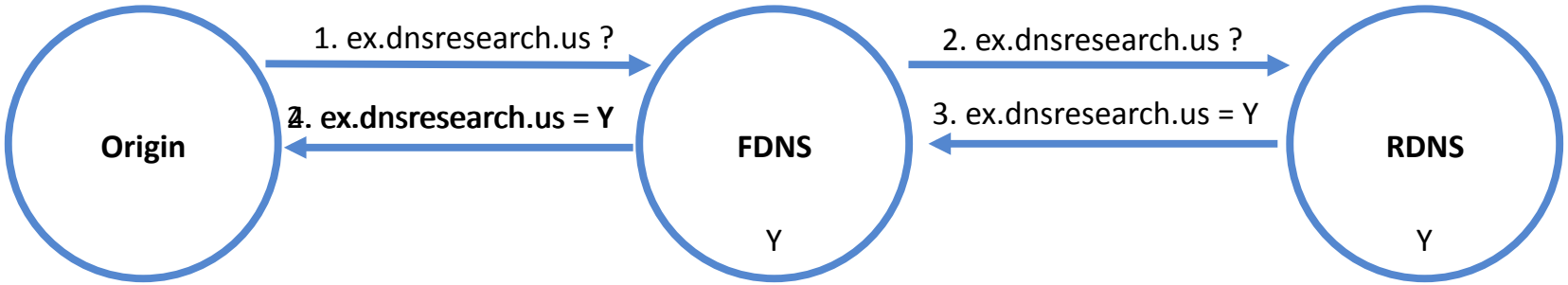
- Records filter through upstream resolvers before arriving at FDNS



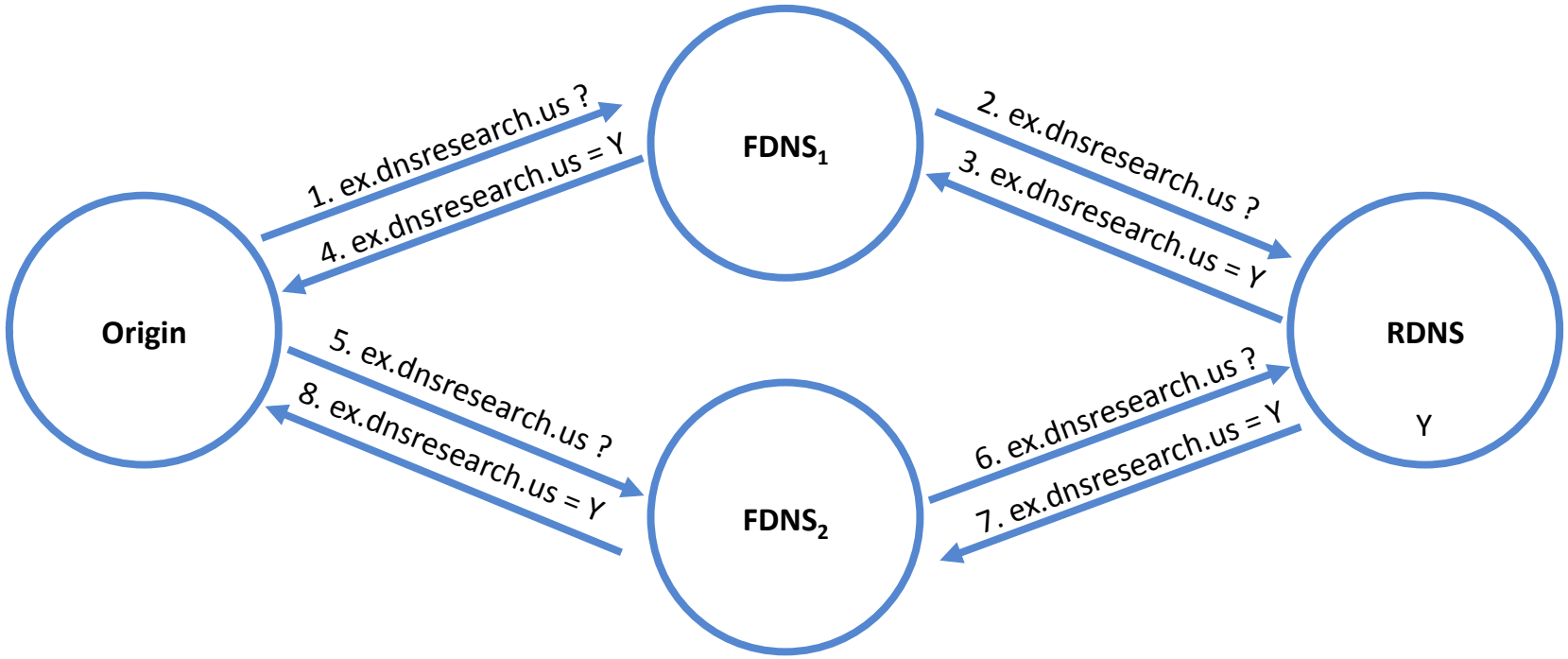
- 7-9% of FDNS vulnerable to cache injection

Measuring RDNS

- Probing an RDNS can be blocked by FDNS caching



Measuring RDNS (Coordinated Probing)

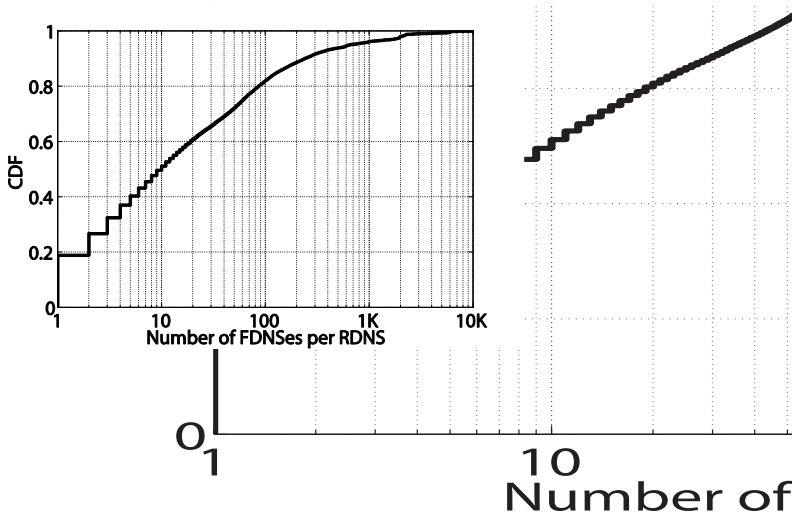


ODNS Population

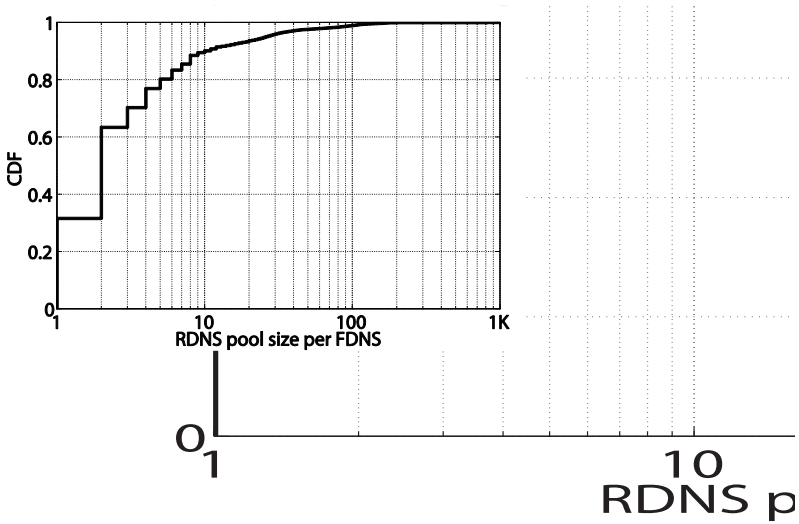
- There are approximately 32 million ODNS
  - Estimation from sampling
- Agrees with full scans from openresolverproject.org
- Previous 2010 study found 15 million ODNS
  - The number of ODNS has doubled within 3 years

FDNS / RDNS Relationship

RDNS are used by many FDNS

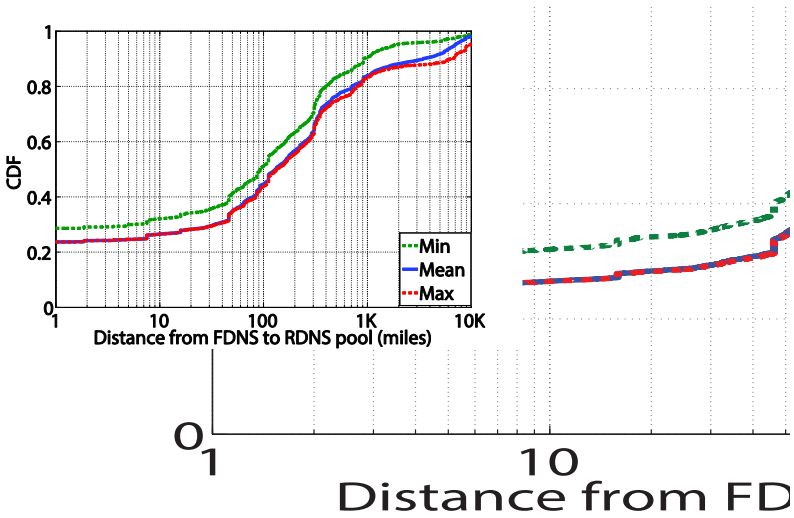


FDNS use "pools" of RDNS resolvers

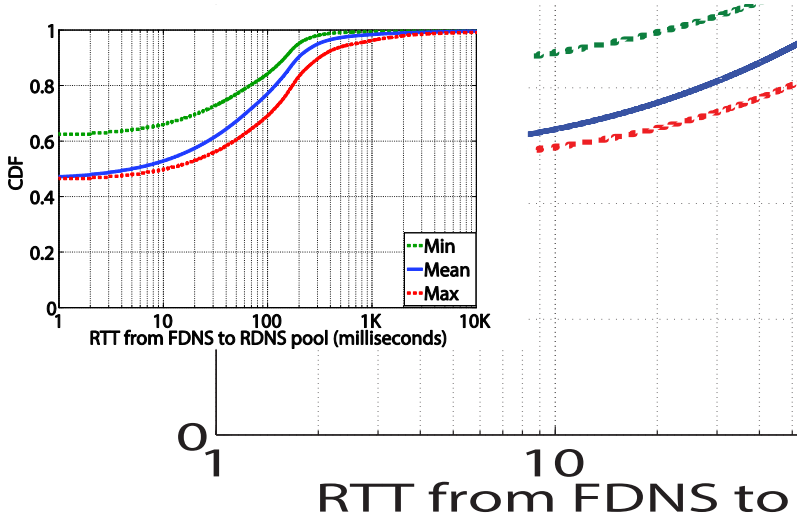


FDNS / RDNS Relationship (cont.)

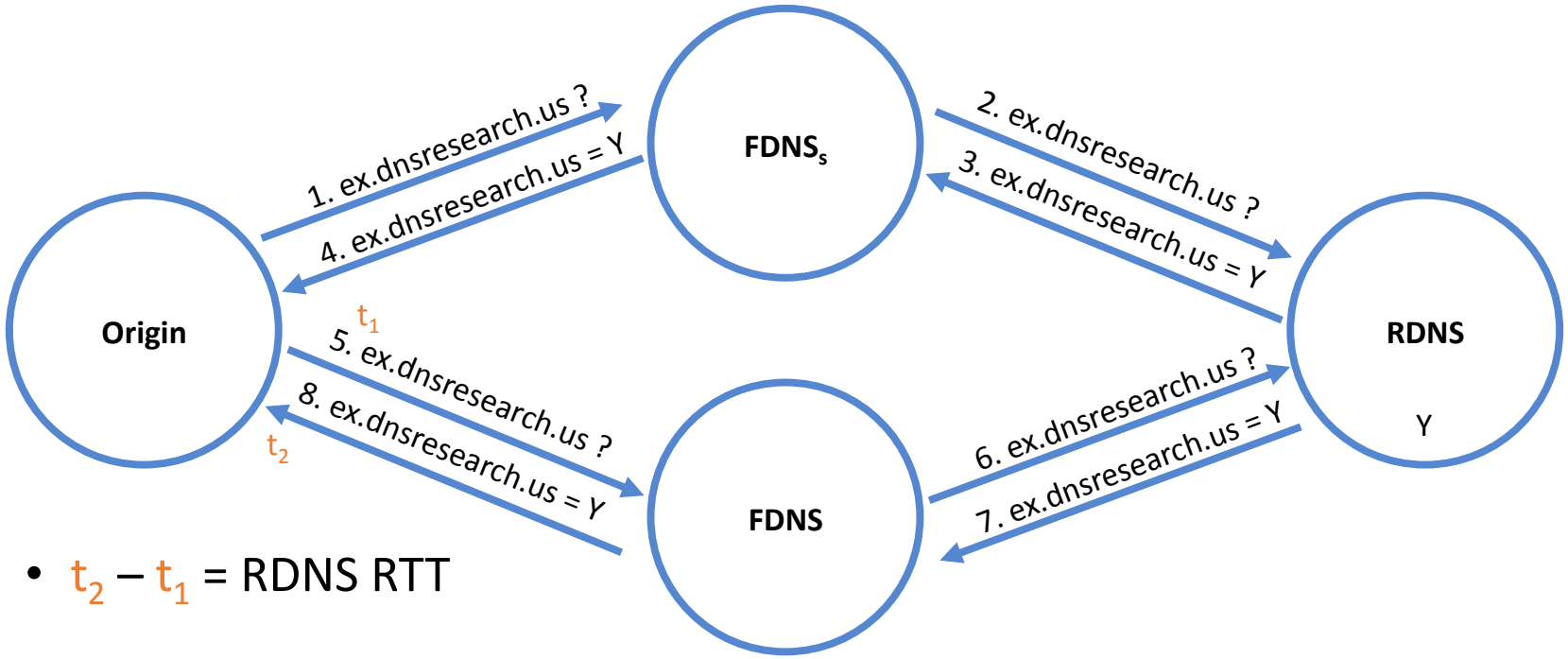
MaxMinds GeoIP database



RTT to RDNS - ICMP ping to FDNS



Measuring RDNS RTT



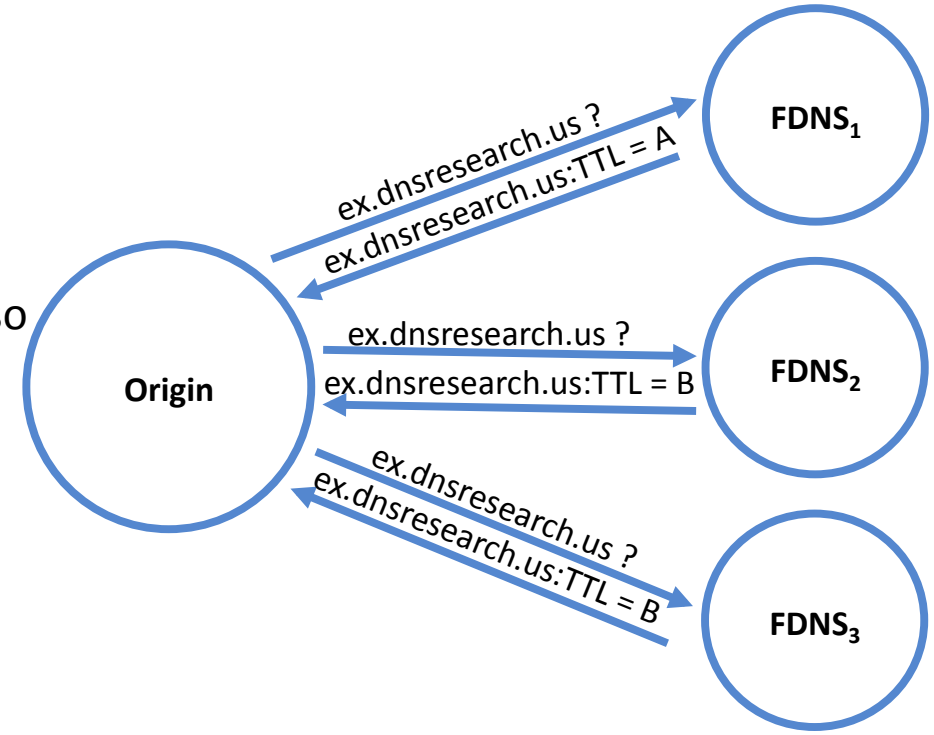
- $t_2 - t_1$ = RDNS RTT

Caching Behavior

- Caching has an important impact on scalability, performance, security
- Example: DNS-based traffic engineering is complicated by caching
  - A single cached DNS record binds an unknown load to the selected server
  - DNS offers a time-to-live (TTL) value to limit the duration of records in cache
  - Many studies have observed that the TTL rule is violated
  - Violations caused by:
    - Resolvers maintaining records in their cache beyond TTL
    - Resolvers modifying the TTL returned to clients

Measuring RDNS TTL Reporting (Voting)

- Expect authoritative TTL X
- Use coordinated probing
- If A == X
  - All actors on path are honest, so
  - RDNS is honest
- Else, majority rule
  - 1 vote for TTL A
  - 2 votes for TTL B – Winner!



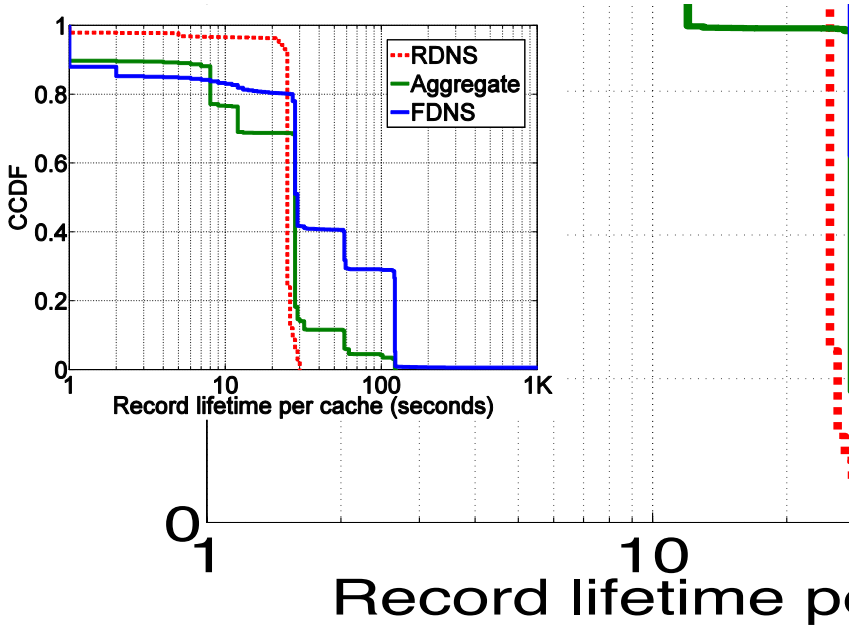
TTL Reporting

- In aggregate, small TTLs are sometimes increased while large TTLs are frequently decreased
- In FDNS, both small and large TTLs are frequently substituted with 10,000 seconds
- In RDNS, small TTLs are rarely misreported while large TTLs are frequently decreased

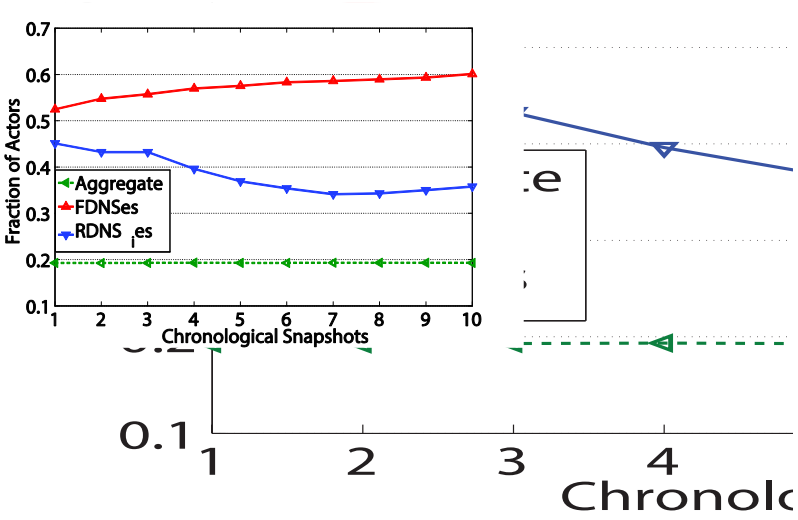
Behavior	Percentage of Measurements		
	Aggregate	FDNS	RDNS
Honest	19%	60%	36%
Lie on Initial	38%	12%	55%
Lie on Subsequent	9%	30%	5%
Constant TTL	7%	26%	5%
Increment TTL	1%	10%	0%

Cache Retention

- Records have a TTL of 30 seconds
- In aggregate, 30% of records are evicted before TTL while 10% are retained for longer than TTL
- In FDNS, 20% of records are evicted before TTL while 40% are retained for longer than TTL
- In RDNS, nearly all records are held for the TTL



Dataset Representativeness



Fraction of actors that honestly report TTL

- Aggregate data is representative
- More "popular" RDNS discovered early in the scan are more likely to be honest
- FDNS dataset is not representative of:
  - All FDNS
  - FDNS that allow cache injection

Conclusion

- We expose the complexity of the client-side DNS infrastructure
  - RDNS pools
  - Multiple layers of resolvers
- There are a significant number of FDNS that are far away from RDNS
- TTL is frequently modified but most often it is reduced
- Records are returned past TTL in only 10% of cases

Thank you! Questions?

Kyle Schomp – kgs7@case.edu

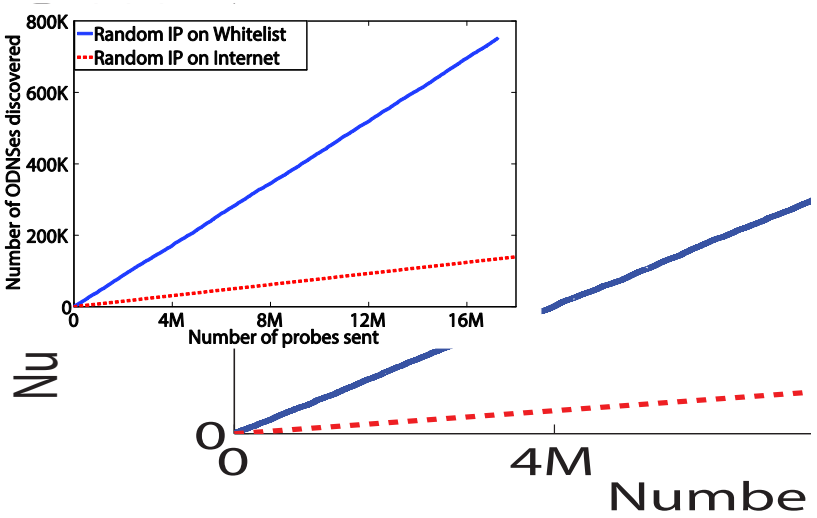
For access to our datasets: http://dns-scans.eecs.cwru.edu/

Additional Slides

Rediscovery

Since ODNS are short-lived, we may need rediscovery

- Scan IP subset twice; second time 3 months after the first
- IP /24 address blocks that were productive tend to remain productive



Datasets

Scan	Format	Start	Dur. (days)	ODNS	RDNS
S_1	Random IP	2/29/12	17	1.09M	69.5K
S_2	Random IP	7/3/12	32	1.98M	72.6K
S_3	Random /24	8/5/12	17	841K	43.9K
S_4	Scan on First Hit	10/4/12	25	17.6M	72.1K
S_5	Rescan of S_3	11/16/12	9	892K	29.9K
S_6	Scan on First Hit	2/26/13	31	11M	65.8K

Residential Network Device Criteria

Criterion	No. ODNSes	% ODNSes
RomPager	258K	24%
Basic auth realm	265K	24%
PBL Listed by SpamHaus	566K	51%
PBL Listed by ISP	180K	17%
Wrong port	529K	48%
Total	849K	78%

TTL Behavior Revisited

Expected (sec)	% <	% >	Mode Lie	
			Value	% of All Lies
1	0%	11%	10000	35%
10-120	<1%	<8%	10000	>37%
1000	1%	3%	10000	62%
3600	2%	2%	10000	51%
10000	5%	0%	3600	40%
10800	8%	0%	3600	27%
86400	16%	0%	21600	36%
100000	22%	0%	21600	27%
604800	22%	0%	21600	26%
1000000	64%	0%	604800	67%

Expected (sec)	% <	% >	Mode Lie	
			Value	% of All Lies
1	0%	31%	10000	88%
10-3600	<1%	19%	10000	>95%
10000	1%	0%	60	92%
10800	19%	0%	10000	97%
86400	19%	0%	10000	97%
100000	19%	0%	10000	97%
604800	19%	0%	10000	97%
1000000	25%	0%	10000	75%

FDNS TTL behavior above and
Aggregate TTL behavior on the left

RDNS TTL Behavior

$RDNS_i$ TTL Behavior

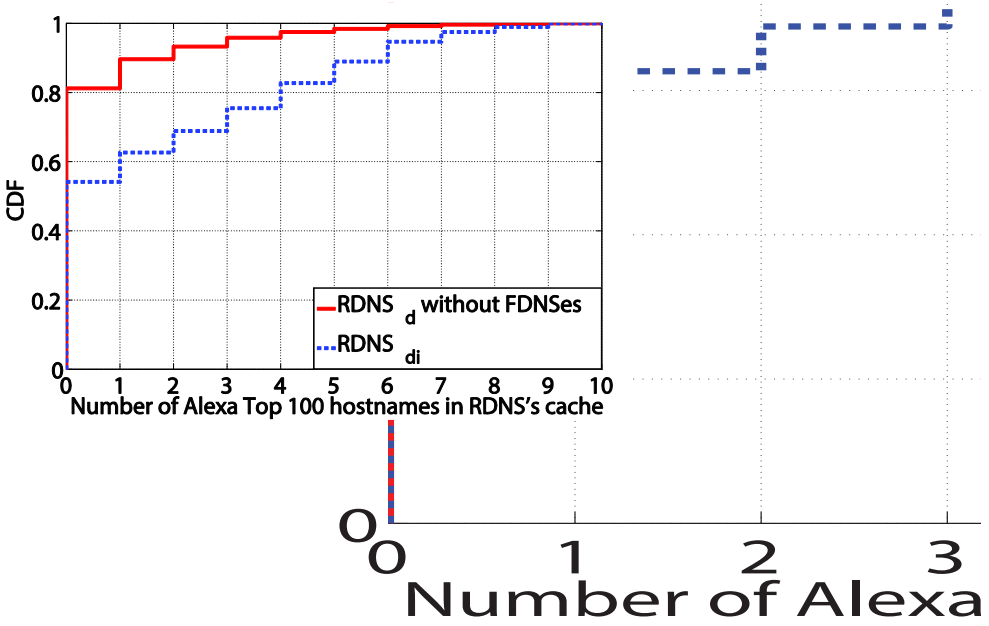
Expected (sec)	% <	% >	Mode Lie	
			Value	% of All Lies
1-120	<1%	<1%	300	>34%
1000	1%	0%	900	29%
3600	1%	0%	80	19%
10000	2%	0%	3600	35%
10800	2%	0%	7200	20%
86400	5%	0%	21600	32%
100000	11%	0%	86400	55%
604800	11%	0%	86400	53%
1000000	49%	0%	604800	71%

$RDNS_{di}$ TTL Behavior

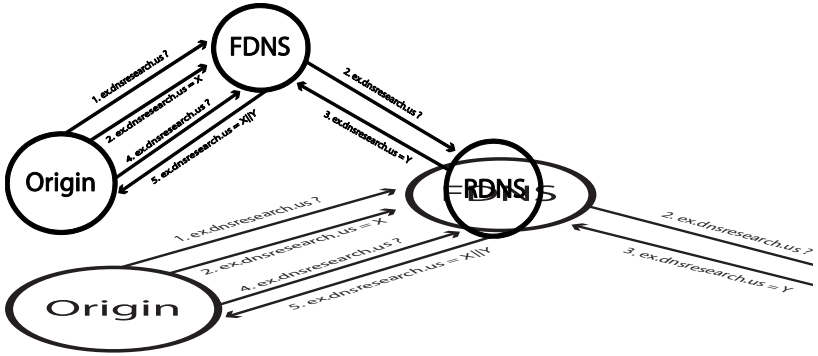
Expected (sec)	% <	% >	Mode Lie	
			Value	% of All Lies
1-120	0%	22%	3600	>52%
1000	3%	19%	3600	53%
3600	3%	7%	86400	69%
10000	16%	7%	3600	53%
10800	16%	7%	3600	52%
86400	16%	0%	3600	72%
100000	40%	0%	86400	59%
604800	40%	0%	86400	59%
1000000	88%	0%	604800	54%

$RDNS_d$ Evaluation

- Both ODNS and RDNS
- Some are not used by any FDNS in the dataset
- What are they? We don't really know
- Since there behavior is different from other RDNS, we opt to remove them from study



Measuring FDNS

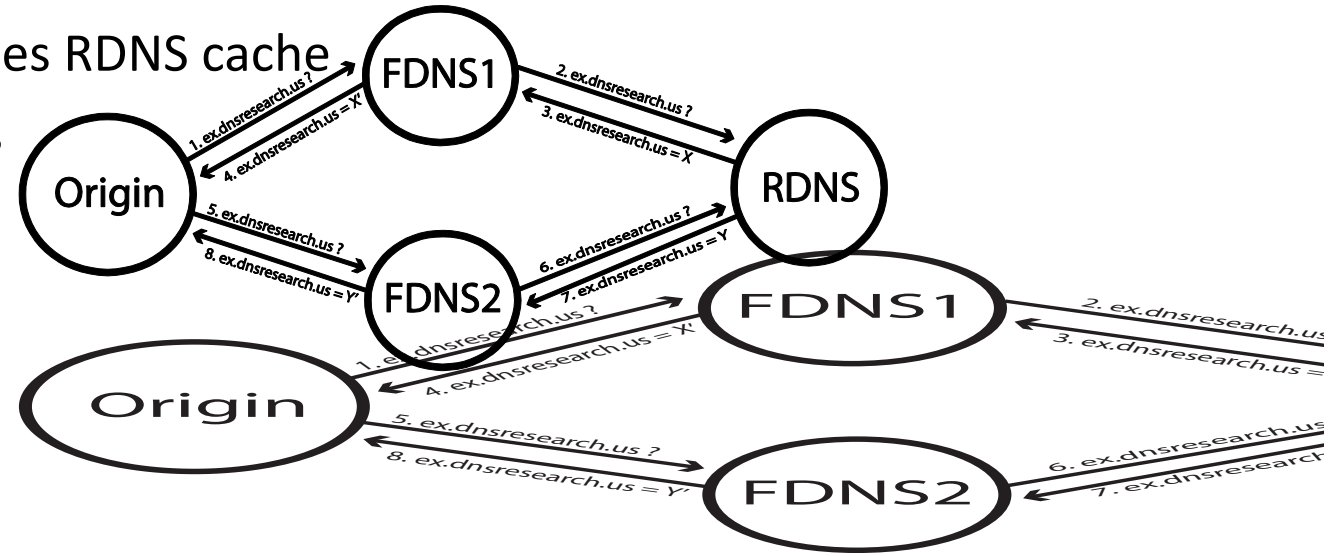


1. Send DNS request to FDNS
2. Immediately send DNS response directly to FDNS binding name to X
3. ADNS response binds name to Y
4. Later, send repeat DNS request to FDNS
5. If response is X, came from FDNS cache

- DNS response from a typical FDNS may come from:
  - FDNS cache
  - HDNS or RDNS cache
  - The ADNS
- 7-9% of FDNS are vulnerable to crude cache poisoning
- They can be measured in isolation

Measuring RDNS

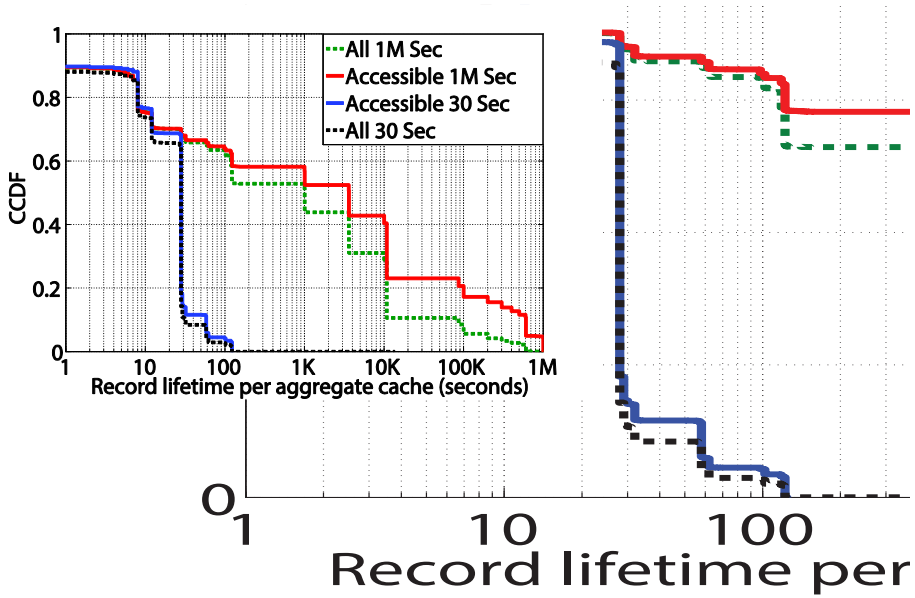
- After a single DNS request, FDNS cache becomes "contaminated"
- FDNS1 primes RDNS cache
- FDNS2 tests
- If X == Y, then the response came from the RDNS



Aggregate Cache Behavior

- Small TTLs are sometimes increased
- Large TTLs are frequently decreased

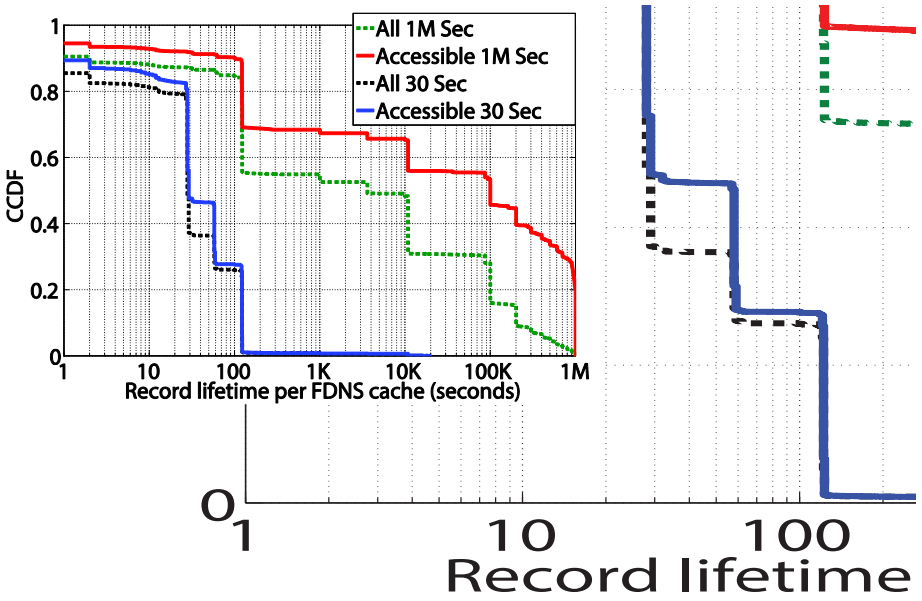
Behavior	Percentage of Measurements
Honest	19%
Lie on Initial	38%
Lie on Subsequent	9%
Constant TTL	7%
Increment TTL	1%



FDNS Cache Behavior

- Both small and large TTLs are frequently substituted with 10,000 seconds
- Not representative of all FDNS

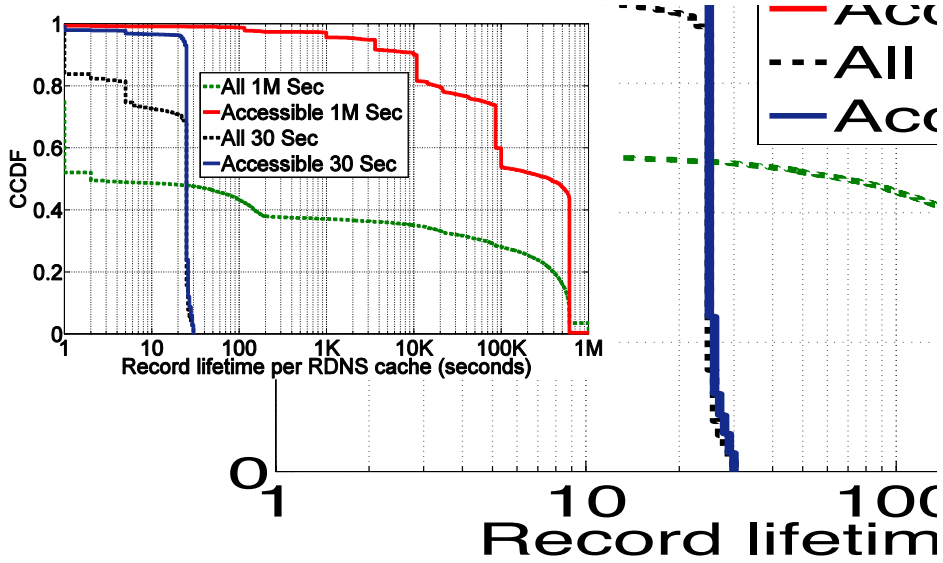
Behavior	Percentage of Measurements
Honest	60%
Lie on Initial	12%
Lie on Subsequent	30%
Constant TTL	26%
Increment TTL	10%



RDNS Cache Behavior

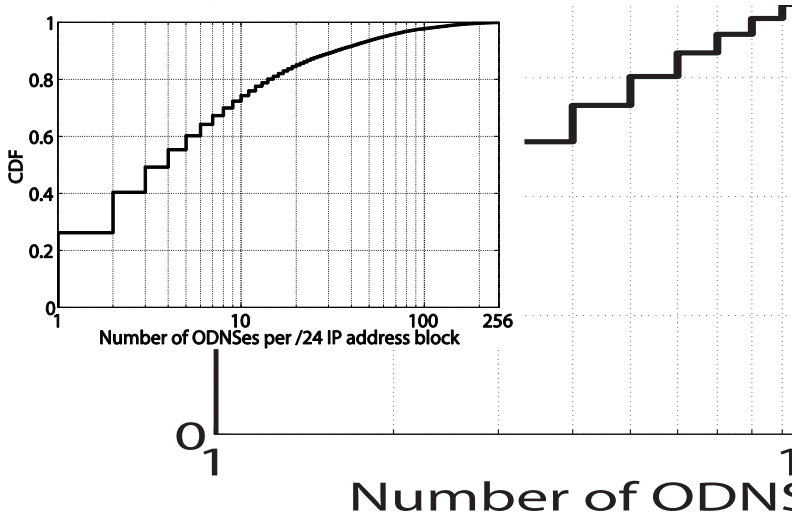
- Small TTLs are rarely misreported
- Large TTLs are frequently decreased

Behavior	Percentage of Measurements
Honest	36%
Lie on Initial	55%
Lie on Subsequent	5%
Constant TTL	5%
Increment TTL	0%

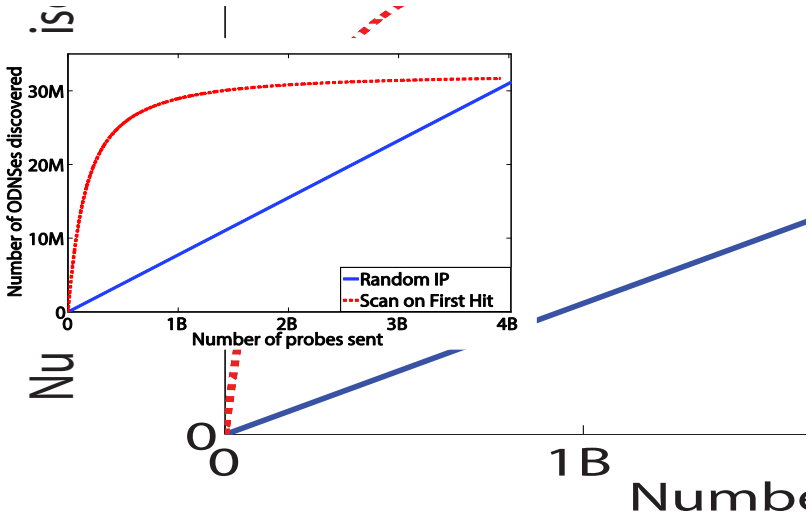


ODNS Discovery

ODNS are unevenly distributed throughout IP space



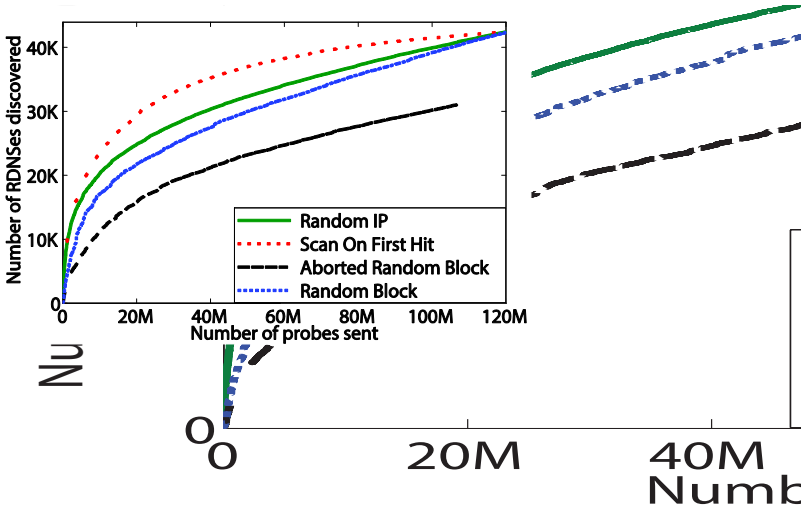
Scan IP addresses randomly vs. Scan /24 IP address block on first ODNS



Extrapolation from a random sample of /24 IP address blocks

RDNS Discovery

- A single FDNS may use many RDNS
  - Send multiple DNS requests to each ODNS
  - CNAME "chain" responses from the ADNS
- New Methodologies
  - Random Block – scan full /24 IP address block
  - Aborted Random Block – stop after discovering first ODNS



Simulation from a random sample of /24 IP address blocks