Inferring Internet Server IPv4 and IPv6 Address Relationships

Robert Beverly, Arthur Berger, Nicholas Weaver, & Larry Campbell

Outline

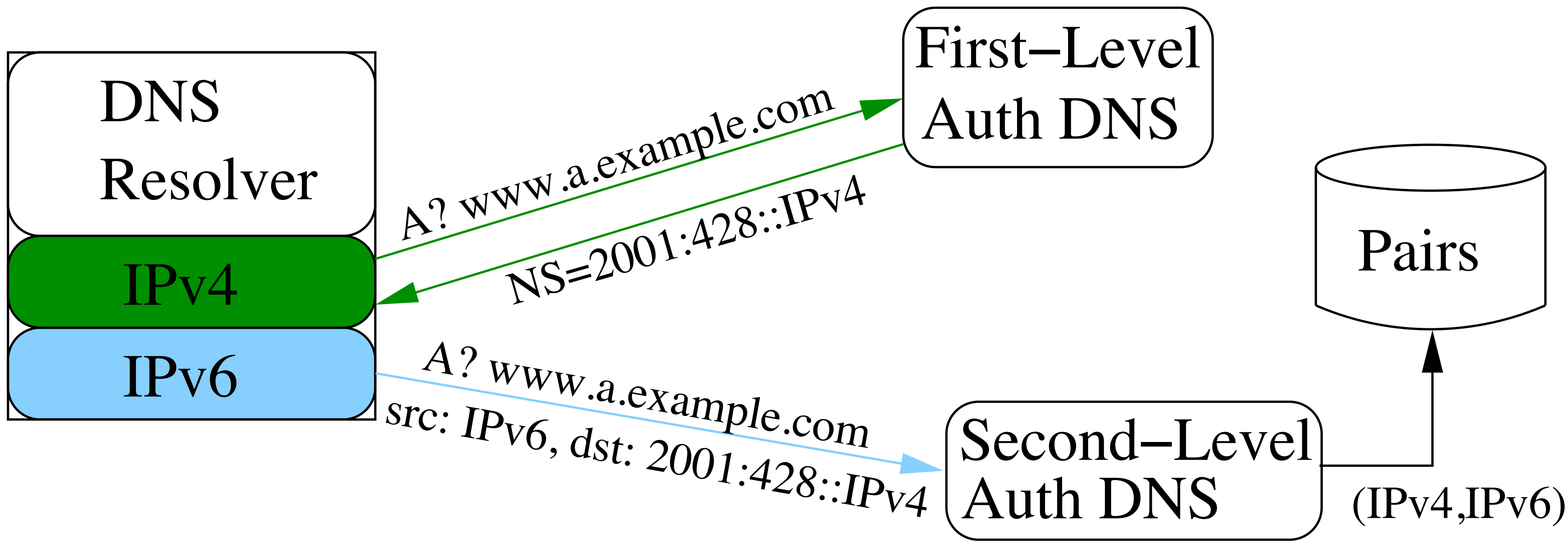
- Introduction
- Opportunistic technique using two-level DNS hierarchy
  - Data set collected by Akamai
- Active probing using a chain of CNAME's
  - Applied to sub-set of Akamai data
- Targeted fingerprinting technique using TCP timestamps
  - Applied to Alexa top 100,000 web servers

Introduction

- Sibling Resolution: Given a candidate (IPv4,IPv6) address pair, determine if these addresses are assigned to the same cluster, device, or interface.

- Why?
  - IPv4 and IPv6 expected to co-exist → dual-stacked devices
  - Track IPv6 evolution
  - Measurements of IPv4 vs. IPv6 performance

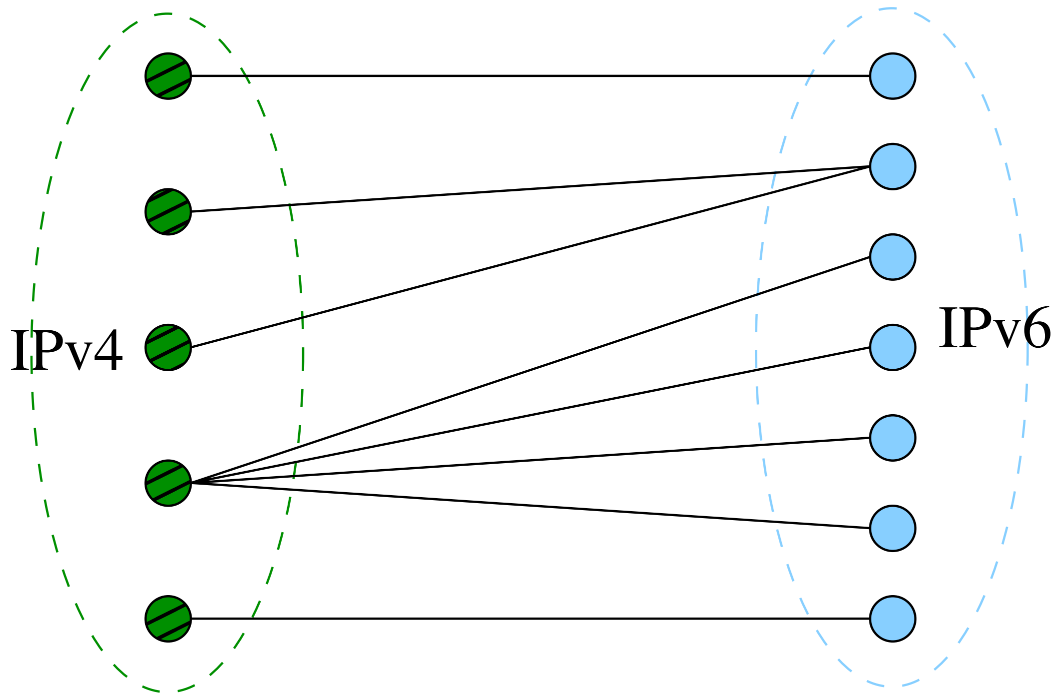
Opportunistic DNS Technique



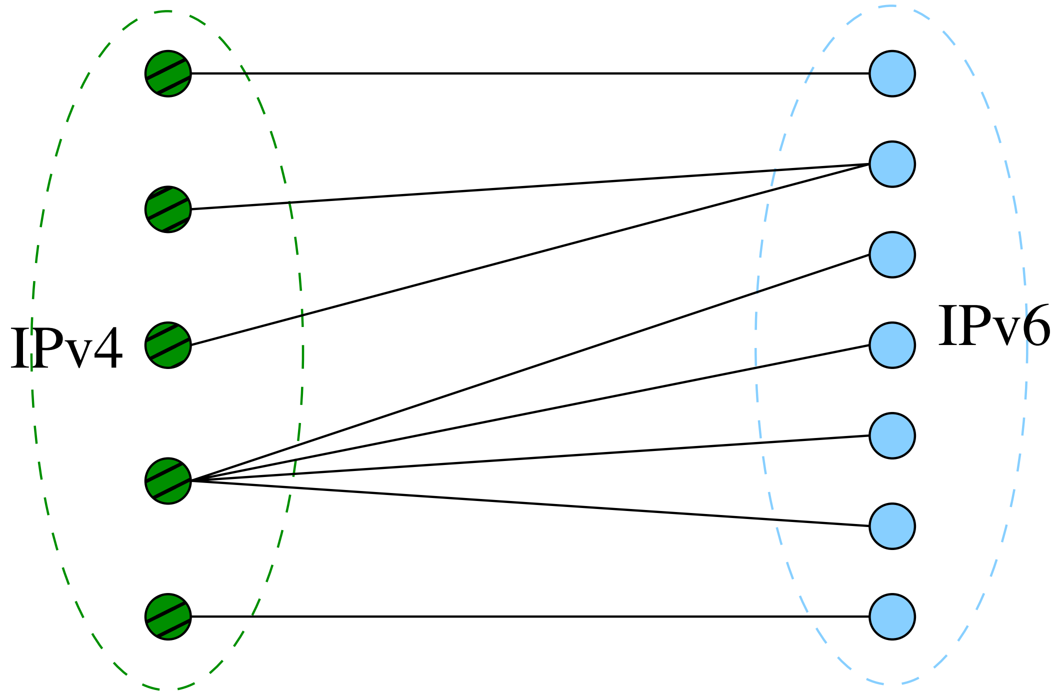
Data Set from Akamai Nameservers

- Six month period from 17 Mar 2012 to 13 Sep 2012.
- 674,000 (v4, v6) pairs.
- 271,000 unique v4 addresses.
- 282,000 unique v6 addresses.
- 213 countries.

Example of Equivalence Classes



Example of Equivalence Classes

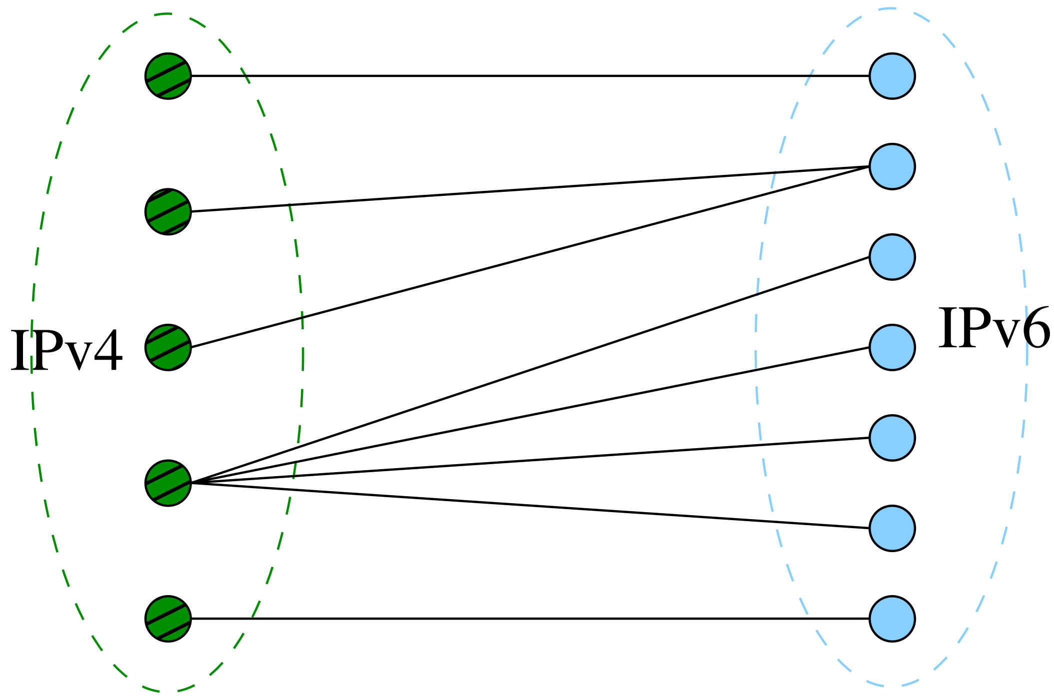


The address pairs partition into 4 equivalence classes:

- two are *1-1*
- one is *2-1*
- one is *1-4*

Will focus first on equivalence classes that are 1-1

Example of Equivalence Classes



- 2 of the 4 equivalence classes (50%) are *1-1*.
- 4 of the 12 addresses (33%) are *1-1.*
- 2 of the 8 address pairs (25%) are *1-1.*

Prevalence of 1-1 equivalence classes

Data Set	Num of pairs	% of eq cls that are 1-1	% of v4+v6 in 1-1 eq cls	% of pairs in 1-1 eq cls
Addresses	674,000	77%	34%	14%
Example	8	50%	33%	25%

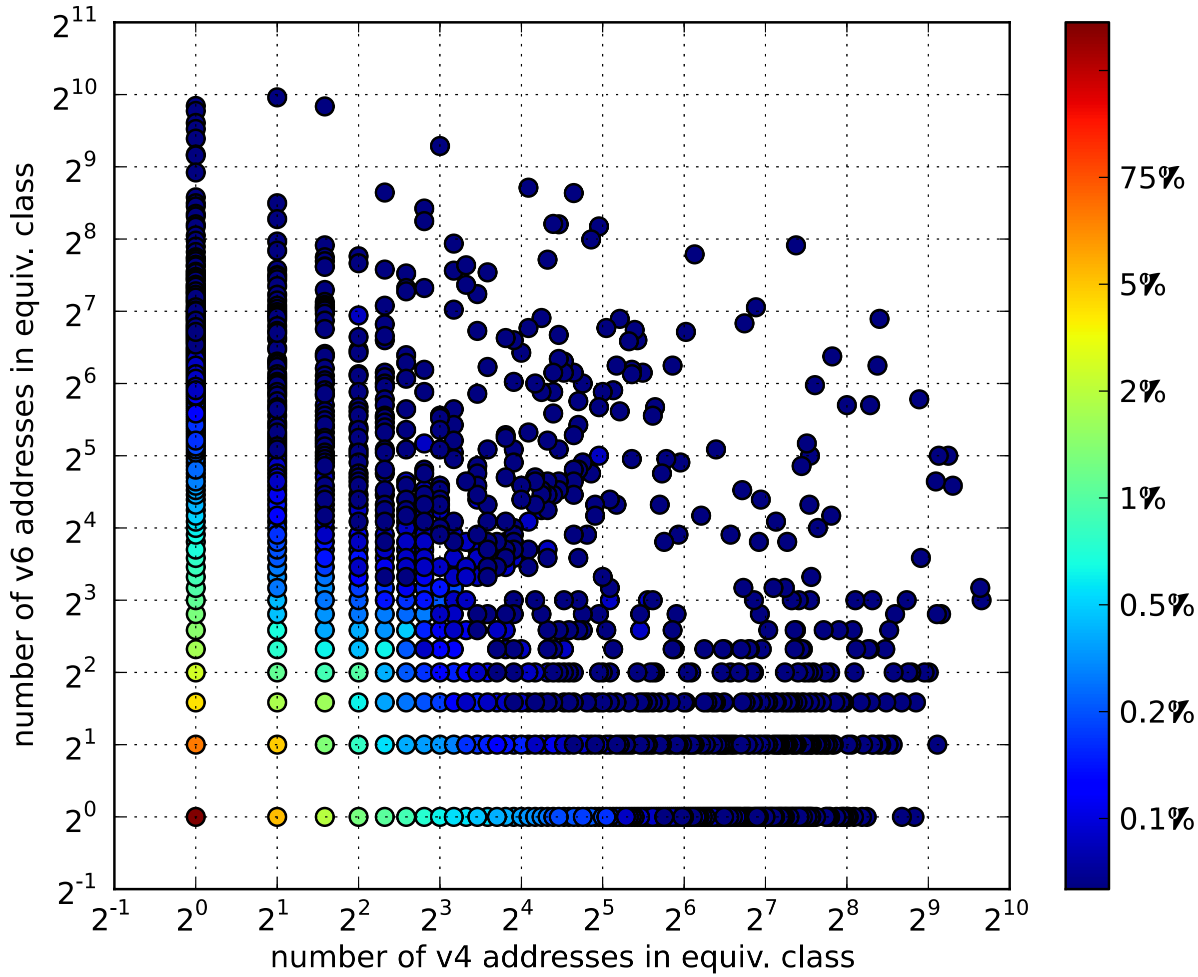
Prevalence of 1-1 equivalence classes

Data Set	Num of pairs	% of eq cls that are 1-1	% of v4+v6 in 1-1 eq cls	% of pairs in 1-1 eq cls
Addresses	674,000	77%	34%	14%
Aggregate to prefixes (before)	238,000	67%	31%	18%
Aggregate to prefixes (after)	260,000	83%	55%	39%
Example	8	50%	33%	25%

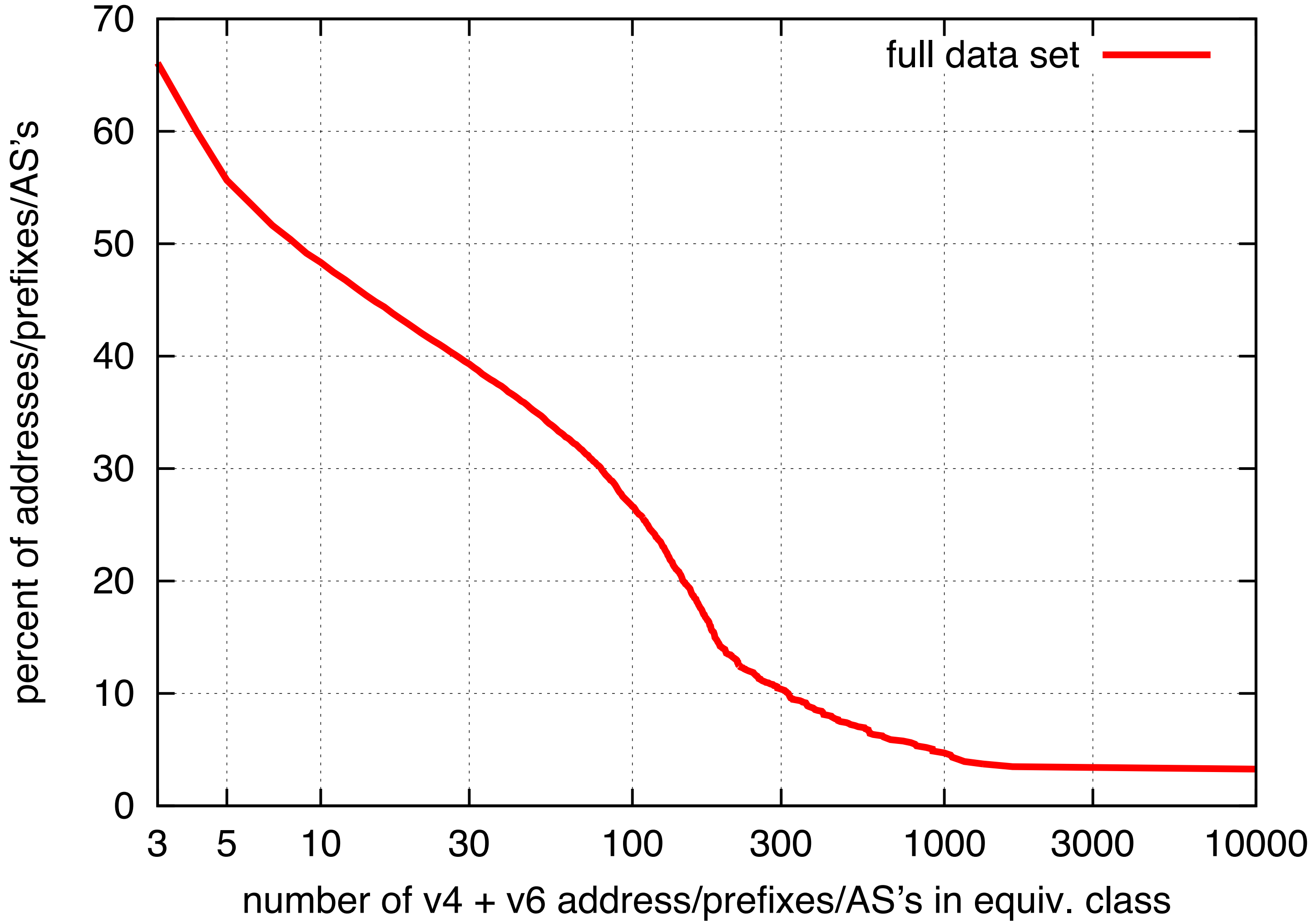
Prevalence of 1-1 equivalence classes

Data Set	Num of pairs	% of eq cls that are 1-1	% of v4+v6 in 1-1 eq cls	% of pairs in 1-1 eq cls
Addresses	674,000	77%	34%	14%
Aggregate to prefixes (before)	238,000	67%	31%	18%
Aggregate to prefixes (after)	260,000	83%	55%	39%
Restrict to last week and aggregate to prefixes (after)	49,000	92%	83%	75%
Aggregate to AS's (after)	55,000	95%	92%	89%
Example	8	50%	33%	25%

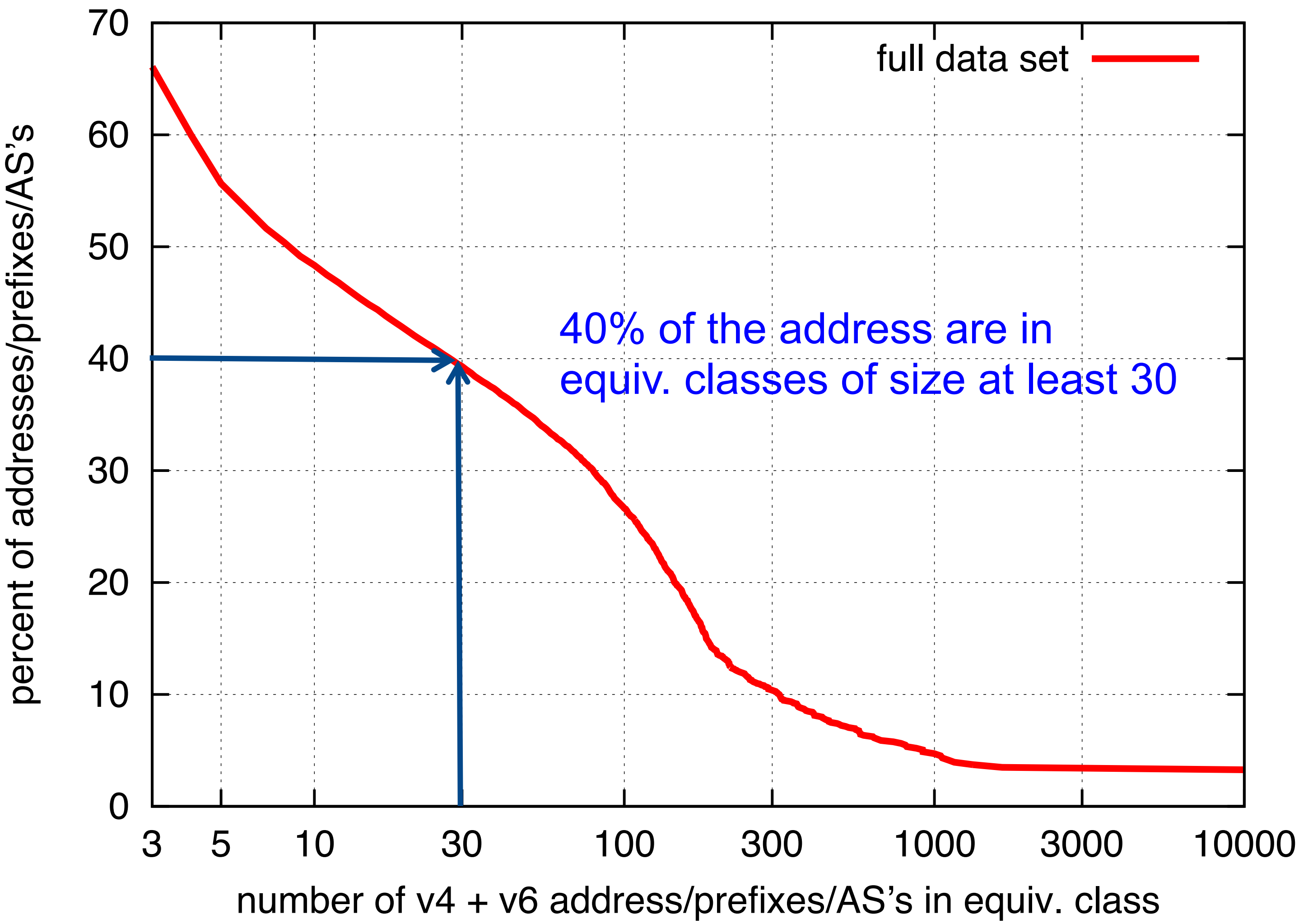
Heat Map of all equivalence classes



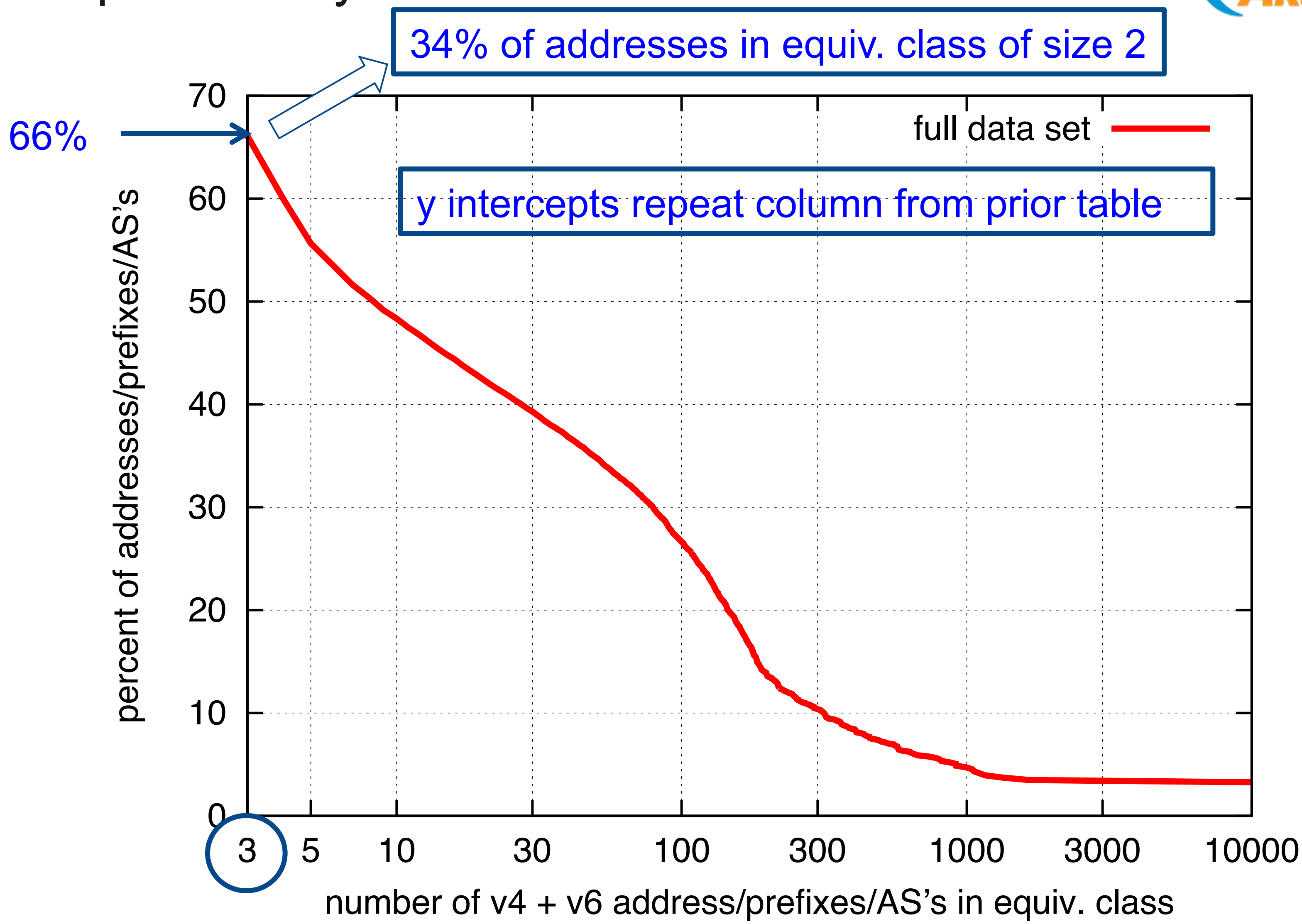
Complementary Distributions



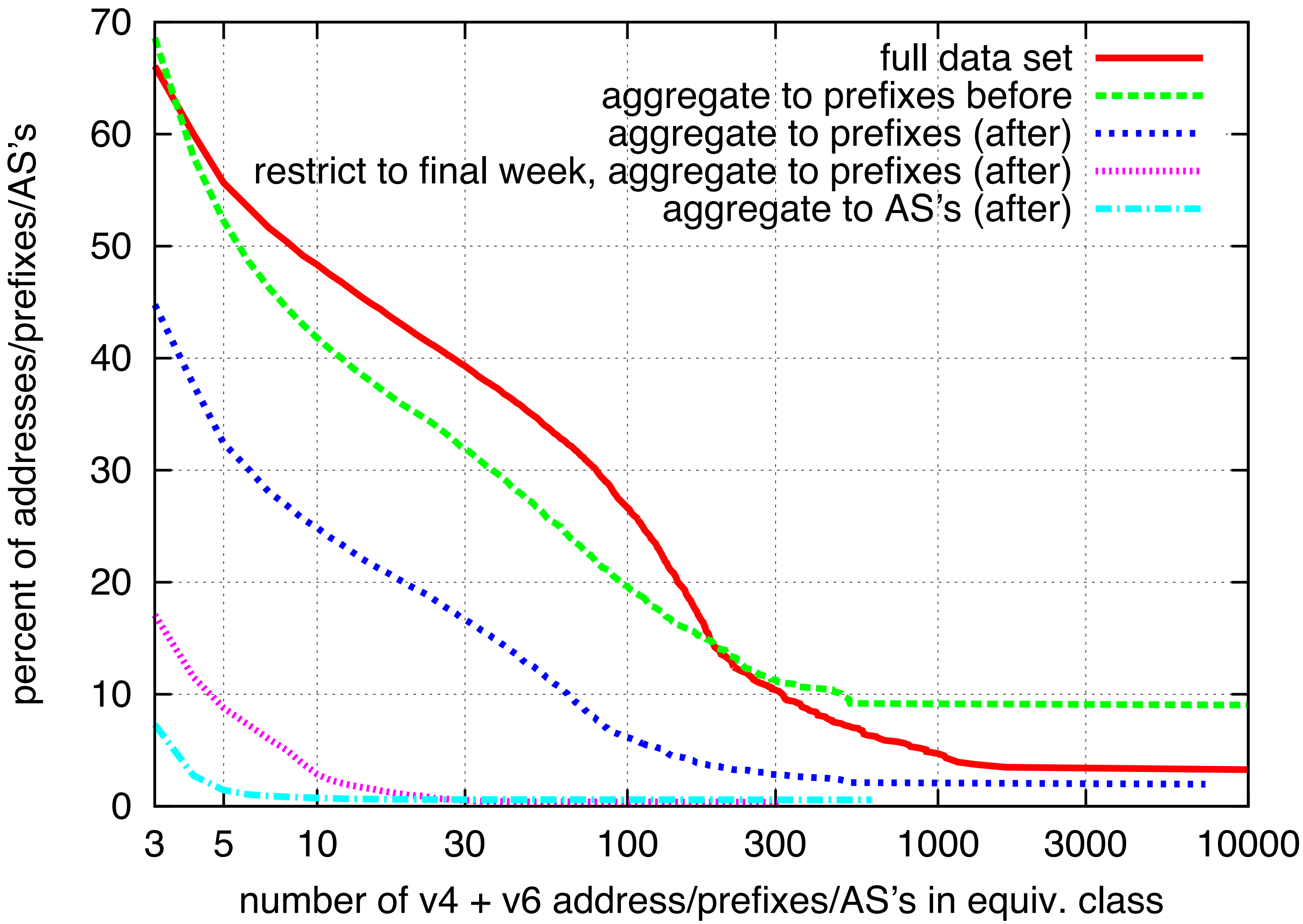
Complementary Distributions



Complementary Distributions



Complementary Distributions



Outline

- Introduction
- Opportunistic technique using two-level DNS hierarchy
  - Data set collected by Akamai
- Active probing using a chain of CNAME's
  - Applied to sub-set of Akamai data
- Targeted fingerprinting technique using TCP timestamps
  - Applied to Alexa top 100,000 web servers

Illustration of Active Probing Technique

dig TXT @8.8.8.8 cname1e6464.nonce.v6.dnstest.icsi.berkeley.edu

Domain controlled by N. Weaver

"

Illustration of Active Probing Technique

dig TXT @8.8.8.8 cname1e6464.nonce.v6.dnstest.icsi.berkeley.edu

The NS record has glue that is only a AAAA

"

Illustration of Active Probing Technique

dig TXT @8.8.8.8 cname1e6464.nonce.v6.dnstest.icsi.berkeley.edu

CNAME

cname2e6464.nonce.2607yf8b0y400dyc02yy16e.v4.dnstest.icsi.berkeley.edu.

"

Encoding of 2607:f8b0:400d:c02::16e

Illustration of Active Probing Technique

dig TXT @8.8.8.8 cname1e6464.nonce.v6.dnstest.icsi.berkeley.edu

CNAME
cname2e6464.nonce.2607yf8b0y400dyc02yy16e.v4.dnstest.icsi.berkeley.edu.

CNAME
cname3e6464.nonce.2607yf8b0y400dyc02yy16e.74x125x176x45.v6.dnstest.icsi.berkeley.edu.

Encoding of 74.125.176.45

Illustration of Active Probing Technique

dig TXT @8.8.8.8 cname1e6464.nonce.v6.dnstest.icsi.berkeley.edu

CNAME

cname2e6464.nonce.2607yf8b0y400dyc02yy16e.v4.dnstest.icsi.berkeley.edu.

CNAME

cname3e6464.nonce.2607yf8b0y400dyc02yy16e.74x125x176x45.v6.dnstest.icsi.berkeley.edu.

CNAME

txt.nonce.2607yf8b0y400dyc02yy16e.74x125x176x45.2607yf8b0y400dyc02yy168.v4.dnstest.icsi.berkeley.edu.

Probe of GoogleDNS anycast address

dig TXT @8.8.8.8 cname1e6464.nonce.v6.dnstest.icsi.berkeley.edu

CNAME
cname2e6464.nonce.2607yf8b0y400dyc02yy16e.v4.dnstest.icsi.berkeley.edu.

CNAME
cname3e6464.nonce.2607yf8b0y400dyc02yy16e.74x125x176x45.v6.dnstest.icsi.berkeley.edu.

CNAME
txt.nonce.2607yf8b0y400dyc02yy16e.74x125x176x45.2607yf8b0y400dyc02yy168.v4.dnstest.icsi.berkeley.edu.

TXT
"nonce" "2607:f8b0:400d:c02::16e" "74.125.176.45" "2607:f8b0:400d:c02::168" "74.125.176.32"

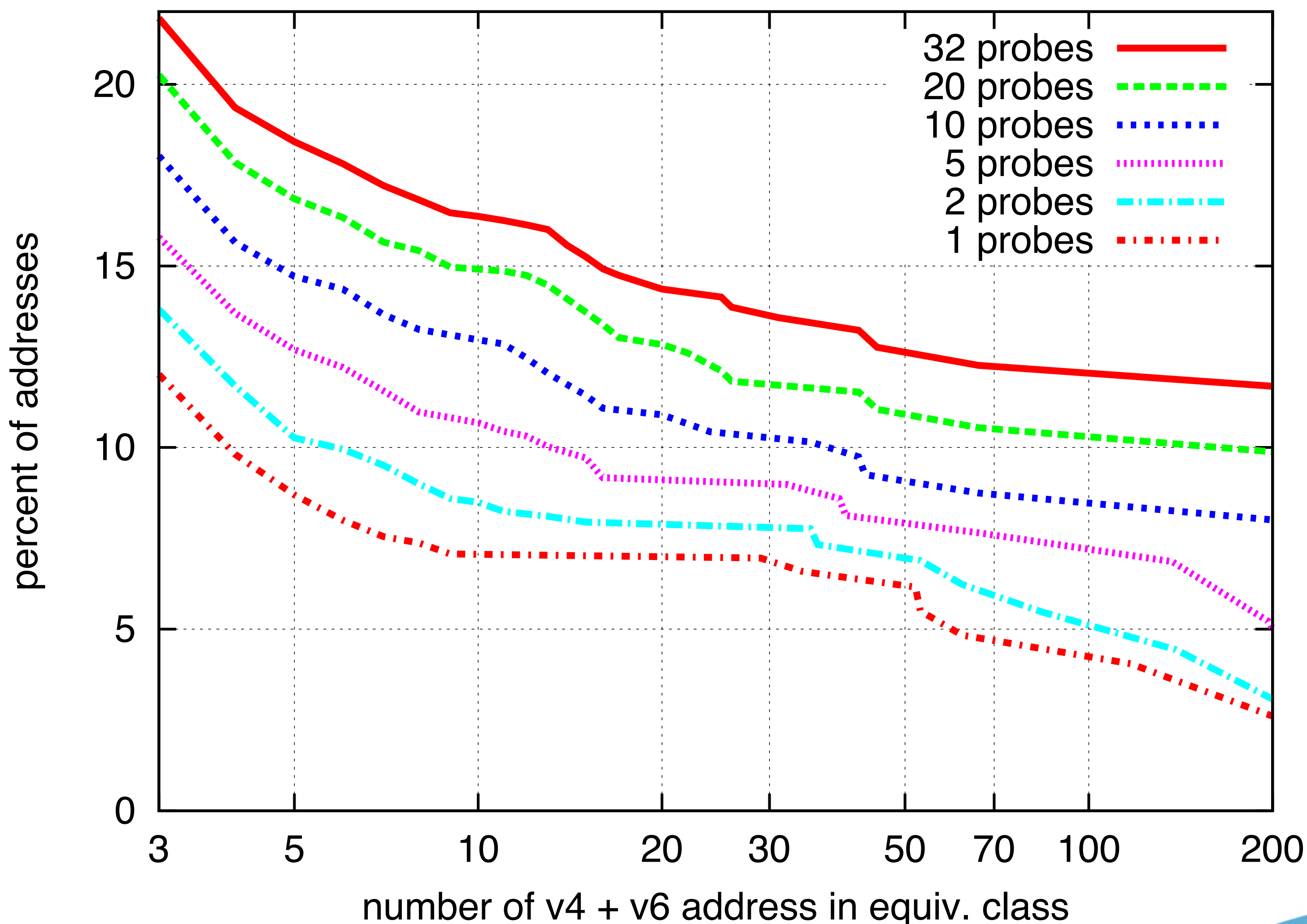
Data Set from Active DNS probing

- Determined the open resolvers in the passive-DNS data set:
  - 6,581 v4 and 2,658 v6 addresses

- Probe each 32 times in 2 hours on Sept 14, 2012.

- Each 4-tuple of v4/v6/v4/v6 yields either 1, 2, or 4 (v4, v6) address pairs.

Complementary distribution of the open resolvers, indexed by number of probes



Outline

- Introduction
- Opportunistic technique using two-level DNS hierarchy
  - Data set collected by Akamai
- Active probing using a chain of CNAME’s
  - Applied to sub-set of Akamai data
- Targeted fingerprinting technique using TCP timestamps
  - Applied to Alexa top 100,000 web servers

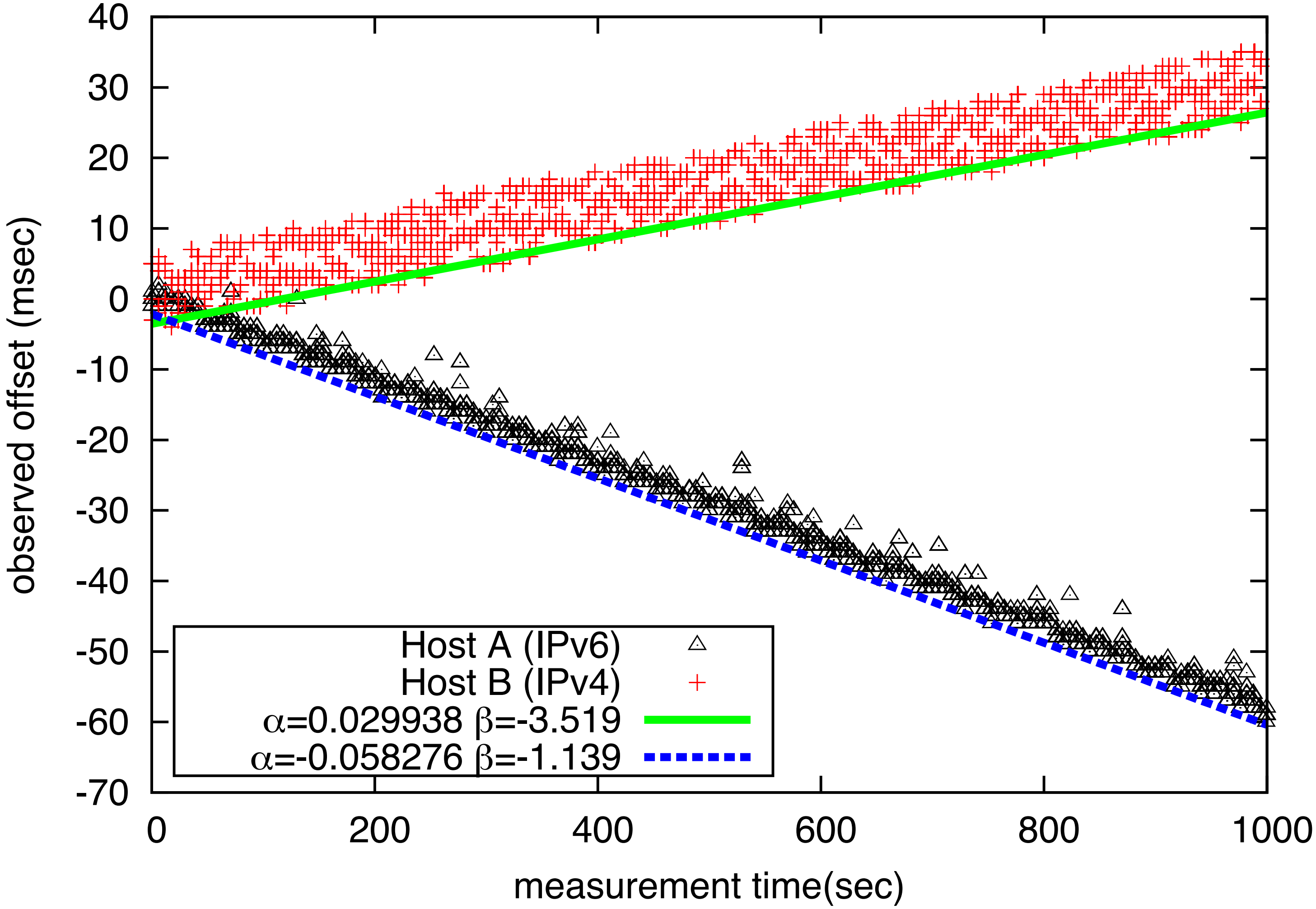
Targeted, Active Technique

- Note that IPv4 and IPv6 share a common transport-layer (TCP) stack.
- Leverage prior work on physical device fingerprinting using TCP timestamp clock skew [Kohno 2005]
- TCP timestamp option: “TCP Extensions for High Performance” [RFC1323, May 1992]
- Widespread support for TCP timestamps (modulo middleboxes, proxies). Enabled by default.

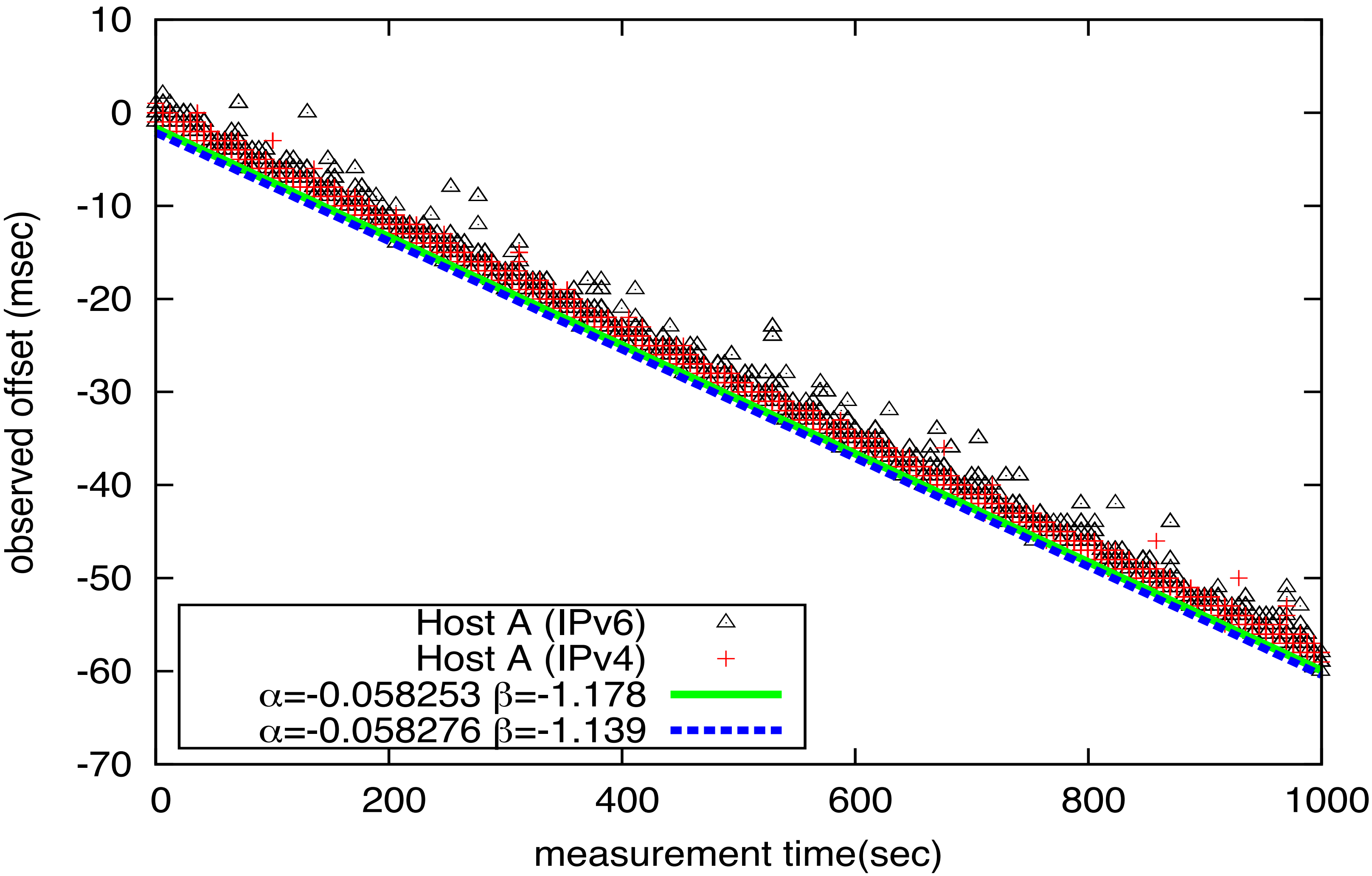
TCP Timestamp Clock Skew

- TS value: 4 bytes with current clock
- TS clock ≠ system clock
- TS clock frequently unaffected by system clock adjustments (e.g. NTP)
- ***Basic Idea:*** Probe over time. Fingerprint is clock *skew* (and remote clock resolution).
- Given a sequence of timestamp offsets, use linear programming to obtain a line that minimizes distance to points, constrained to be under data points. [Moon, 1999]

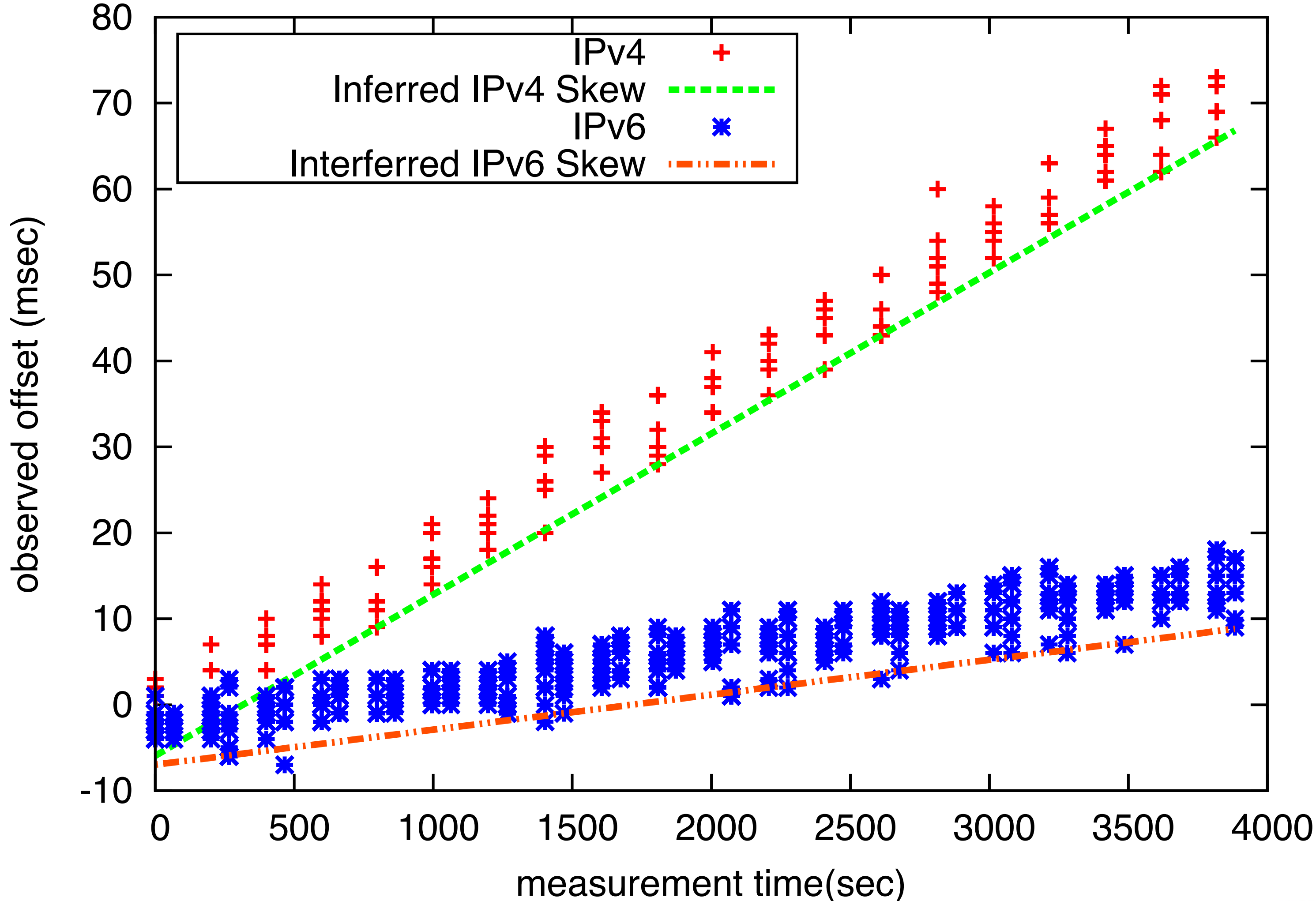
Control test on known **distinct** machines



Control test on known **common** machine



Inferred clock skew to www.socialsecurity.gov



Sibling Inference at Alexa Websites

- Analyze Alexa top 100,000 websites
- Pull `A` and `AAAA` records
- 1398 (1.4%) have IPv6 DNS
- Repeatedly fetch root HTML page via IPv4 and IPv6 via deterministic IP address
- Record all packets
- Infer siblings if the angle between the two fitted lines is within 1 degree.

Sibling Inference at Alexa Websites

Case	Inference	Count
v4 and v6 no timestamps	?	94 (6.7%)
v4 or v6 (but not both) no timestamps	Non-siblings	101 (7.2%
v4 and v6 non-monotonic	?	109 (7.8%)
v4 or v6 (but not both) non-monotonic	Non-siblings	140 (10.0%)

- Our technique fails when timestamps are not monotonic across TCP flows (e.g. load-balancer or BSD OS)
- Or, when timestamps are not supported (e.g. middlebox)
- But when this occurs for just one of the addresses, can infer non-siblings

Sibling Inference at Alexa Websites

Case	Inference	Count
v4 and v6 no timestamps	?	94 (6.7%)
v4 or v6 (but not both) no timestamps	Non-siblings	101 (7.2%
v4 and v6 non-monotonic	?	109 (7.8%)
v4 or v6 (but not both) non-monotonic	Non-siblings	140 (10.0%)
Clock skew satisfies criterion	Siblings	839 (60.0%)
Clock skew fails criterion	Non-siblings	115 (8.3%)
Total		1398 (100%)

- 25.5% (356) non-siblings
- 43% of skew-based non-siblings are in different ASes

Summary: Characterizing the inter-relation of v4 and v6 among Internet DNS and web servers.

Presented three methodologies:

1. a passive DNS collection using a two-level DNS hierarchy
2. an active DNS probing system using a chain of CNAME's, and can force resolvers to utilize TCP
3. an active TCP physical device fingerprinting technique that more precisely identifies v4 and v6 addresses present on the same machine.

Summary: Characterizing the inter-relation of v4 and v6 among Internet DNS and web servers.

Presented three methodologies:

1. a passive DNS collection using a two-level DNS hierarchy
2. an active DNS probing system using a chain of CNAME’s, and can force resolvers to utilize TCP
3. an active TCP physical device fingerprinting technique that more precisely identifies v4 and v6 addresses present on the same machine.

We find:

1. significant complexity, as measured by large equivalence classes.
2. 25% of the top Alexa sites that resolve to `A` and `AAAA` are non-siblings.

people.csail.mit.edu/awberger/papers/v4_v6_address_relationships.pdf

Additional Slides

Illustration:

```
$ dig +trace +additional a10.dspg1.akamai.net
```

Additional Section from First Level Nameserver:

```
n0dspg1.akamai.net. 21600     IN   A    195.59.43.138
a0dspg1.akamai.net. 32400     IN   AAAA 2001:5000:402:f000:2b85:c412:8ce4:c418
n5dspg1.akamai.net. 32400     IN   A    23.3.10.154
n3dspg1.akamai.net. 43200     IN   A    23.3.10.150
a1dspg1.akamai.net. 21600     IN   AAAA 2001:218:2007:ffff:9208:c412:8ce4:c418
n2dspg1.akamai.net. 32400     IN   A    193.108.88.193
n1dspg1.akamai.net. 43200     IN   A    61.213.146.8
n4dspg1.akamai.net. 21600     IN   A    66.171.230.14
```

Encodes the IPv4 source address of the incoming DNS query

Illustration:

```
$ dig +trace +additional a10.dspg1.akamai.net
```

Additional Section from First Level:

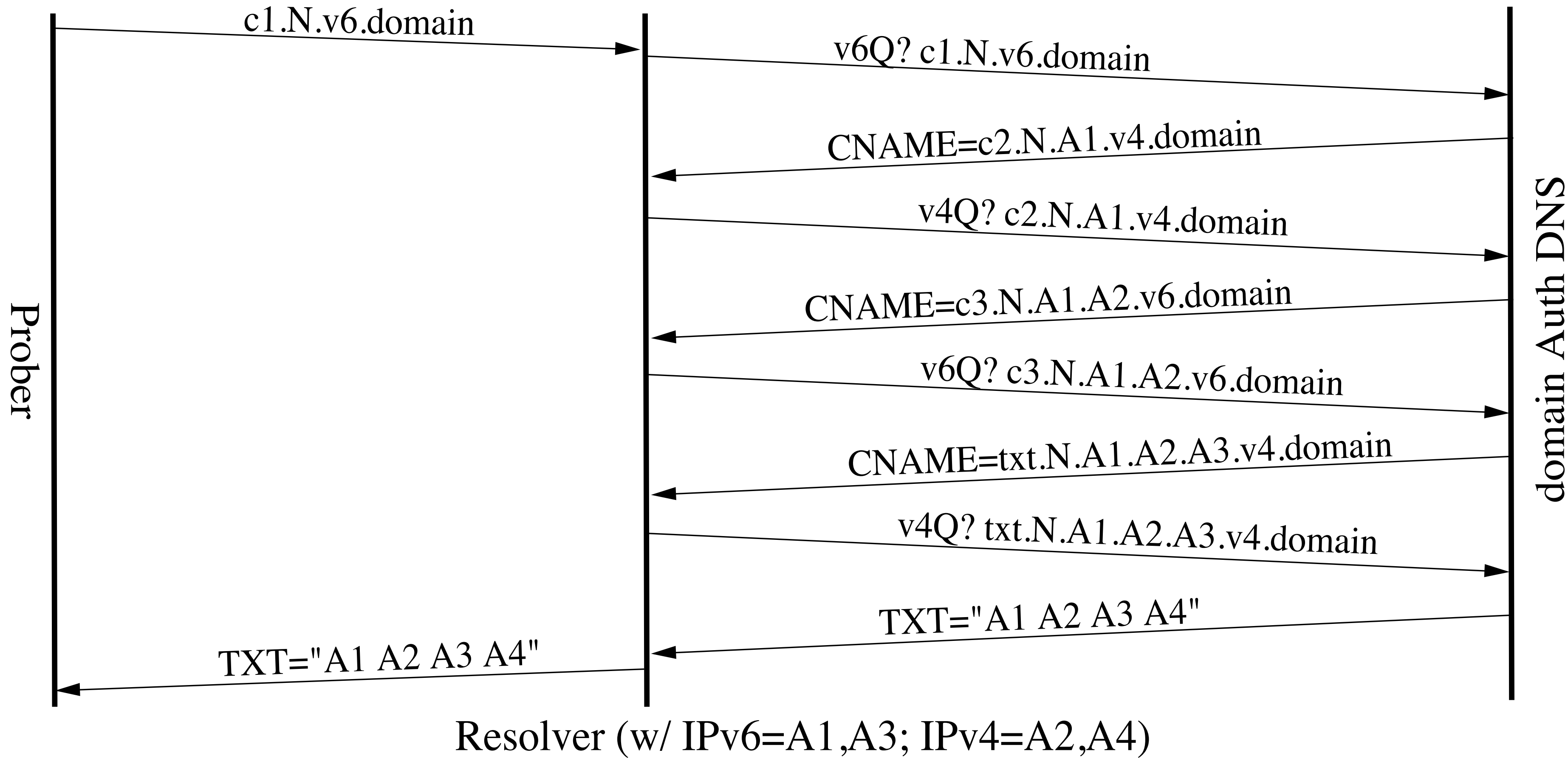
```
n0dspg1.akamai.net. 21600     IN   A    195.59.43.138
a0dspg1.akamai.net. 32400     IN   AAAA 2001:5000:402:f000:2b85:c412:8ce4:c418
n5dspg1.akamai.net. 32400     IN   A    23.3.10.154
n3dspg1.akamai.net. 43200     IN   A    23.3.10.150
a1dspg1.akamai.net. 21600     IN   AAAA 2001:218:2007:ffff:9208:c412:8ce4:c418
n2dspg1.akamai.net. 32400     IN   A    193.108.88.193
n1dspg1.akamai.net. 43200     IN   A    61.213.146.8
n4dspg1.akamai.net. 21600     IN   A    66.171.230.14
```

Encodes the IPv4 source address of the incoming DNS query

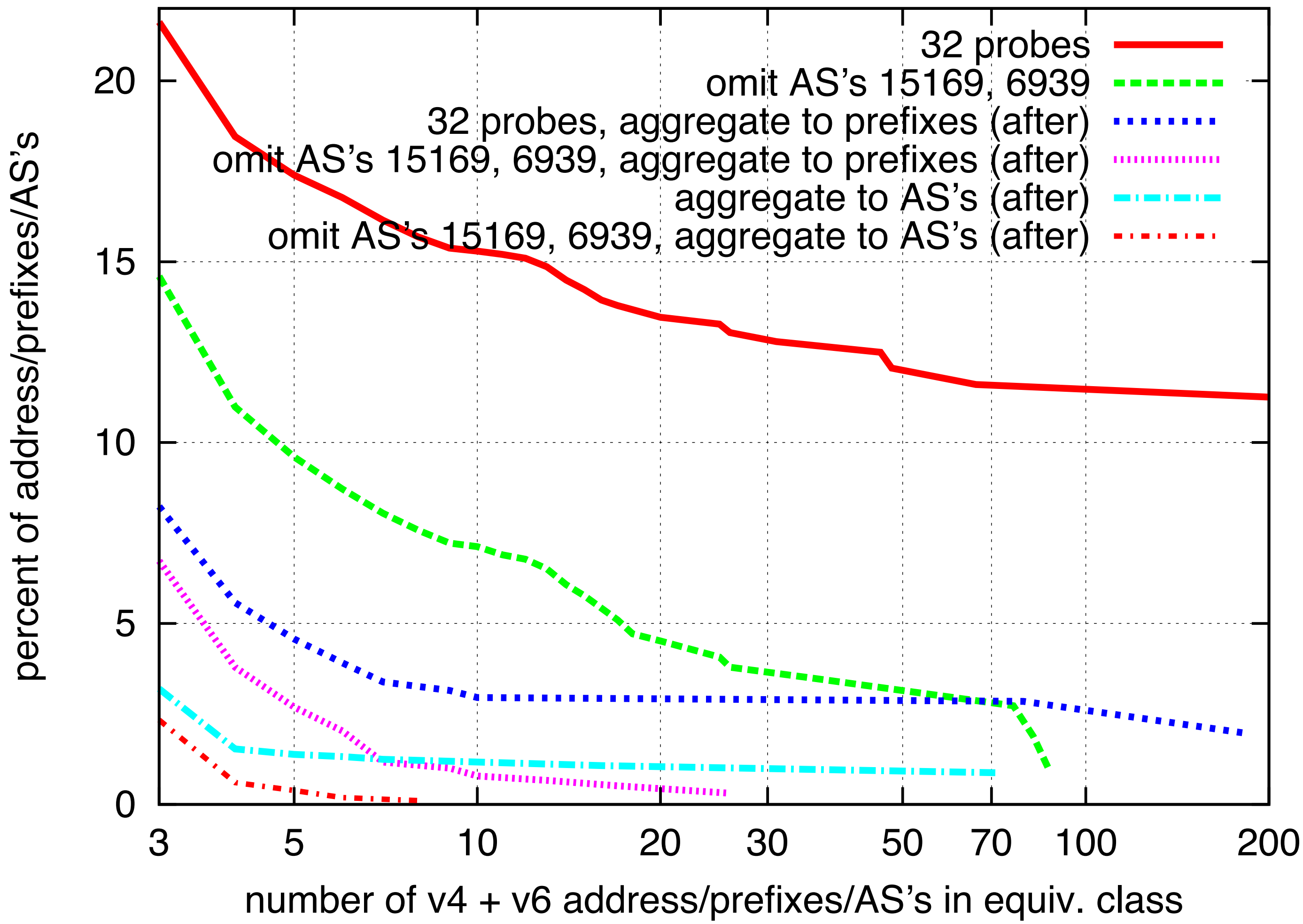
```
Resolution of domain:
a10.dspg1.akamai.net.    20   IN   A    80.67.64.115
a10.dspg1.akamai.net.    20   IN   A    80.67.64.116
```

Note: protocol version of answer is independent of that used to transport the DNS messages

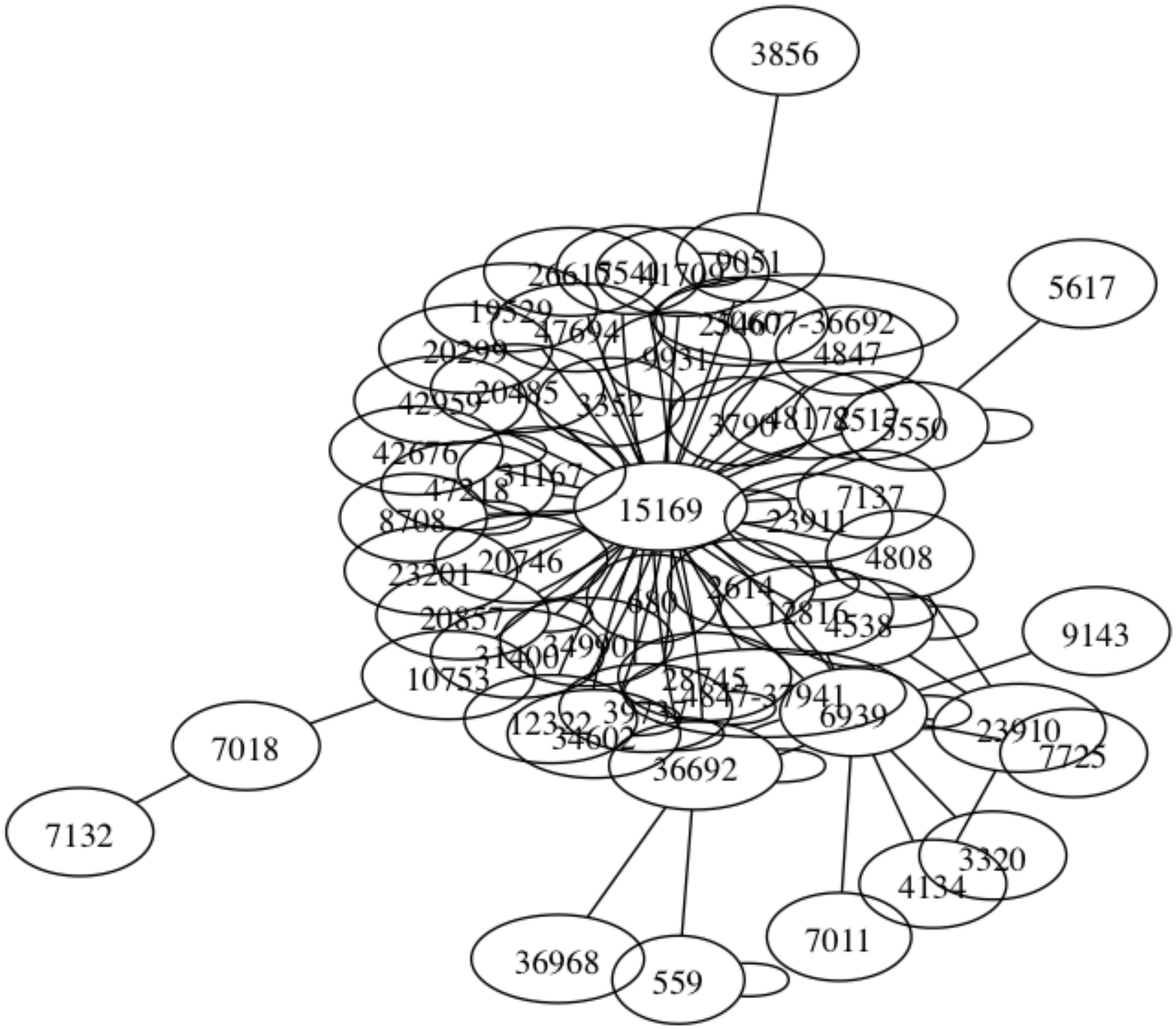
Active probing to open, recursive resolvers using a chain of CNAME's



Complementary distribution of the open resolvers



Graph of largest equiv. class, aggregated to AS's.



Timestamp offsets

Let t_i be the time at which the prober observes the i^{th} v4 packet

Let T_i be the timestamp in the TCP options of the i^{th} v4 packet.

Then the offset of the i^{th} v4 packet = $(T_i - T_1) - (t_i - t_1)$

Likewise for the v6 packets.

Timestamp offsets

Let t_i be the time at which the prober observes the i^{th} v4 packet

Let T_i be the timestamp in the TCP options of the i^{th} v4 packet.

Then the offset of the i^{th} v4 packet = $(T_i - T_1) - (t_i - t_1)$

Likewise for the v6 packets.

Given a sequence of timestamp offsets, use linear programming to obtain a line that minimizes distance to points, constrained to be under data points. [Moon, 1999]