Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure via Active Fingerprinting

Robert Beverly*, Arthur Berger†

*Naval Postgraduate School
†MIT/Akamai

March 20, 2015

PAM 2015 - 16th Passive and Active Measurement Conference

Outline

IPv4/IPv6 Siblings

IPv4/IPv6 "Siblings:"

Given a candidate (*IPv4*, *IPv6*) address pair, determine if these addresses are assigned to the same physical machine.

Related IPv6 Research:

- IPv6 adoption, routing, performance [DLHEA12], [CAZIOB14]
- Passive client IPv4/IPv6 sibling associations: e.g. web-bugs, javascript, flash [ZAAHM12]
- DNS server IPv4/IPv6 siblings [BWBC13]

Our work:

- *Targeted, active test:* on-demand for any given pair
- *Infrastructure:* finding server siblings

IPv4/IPv6 Siblings

IPv4/IPv6 "Siblings:"

Given a candidate (*IPv4*, *IPv6*) address pair, determine if these addresses are assigned to the same physical machine.

Related IPv6 Research:

- IPv6 adoption, routing, performance [DLHEA12], [CAZIOB14]
- Passive client IPv4/IPv6 sibling associations: e.g. web-bugs, javascript, flash [ZAAHM12]
- DNS server IPv4/IPv6 siblings [BWBC13]

Our work:

- *Targeted, active test:* on-demand for any given pair
- *Infrastructure:* finding server siblings

Motivation

Question?

Is IPv6 infrastructure being deployed with separate hardware or by adding IPv6 to existing machines?

Why?

- ***Adoption:***
  - Track IPv6 infrastructure evolution, how deployed
- ***Bootstrapping:***
  - IPv6 geolocation, reputation by correlating to IPv4 counterpart
- ***Security:***
  - Better understand correlated failures
  - Lack of IPv6 security, tunnel to circumvent firewalls
  - (e.g. an attack on IPv6 resource affecting IPv4 service)
- ***Performance:***
  - Isolate path vs. host performance when comparing IPv4 and IPv6

Motivation

Question?

Is IPv6 infrastructure being deployed with separate hardware or by adding IPv6 to existing machines?

Why?

- ***Adoption:***
  - Track IPv6 infrastructure evolution, how deployed
- ***Bootstrapping:***
  - IPv6 geolocation, reputation by correlating to IPv4 counterpart
- ***Security:***
  - Better understand correlated failures
  - Lack of IPv6 security, tunnel to circumvent firewalls
  - (e.g. an attack on IPv6 resource affecting IPv4 service)
- ***Performance:***
  - Isolate path vs. host performance when comparing IPv4 and IPv6

Motivation

Question?

Is IPv6 infrastructure being deployed with separate hardware or by adding IPv6 to existing machines?

Why?

- ***Adoption:***
  - Track IPv6 infrastructure evolution, how deployed
- ***Bootstrapping:***
  - IPv6 geolocation, reputation by correlating to IPv4 counterpart
- ***Security:***
  - Better understand correlated failures
  - Lack of IPv6 security, tunnel to circumvent firewalls
  - (e.g. an attack on IPv6 resource affecting IPv4 service)
- ***Performance:***
  - Isolate path vs. host performance when comparing IPv4 and IPv6

Motivation

Question?

Is IPv6 infrastructure being deployed with separate hardware or by adding IPv6 to existing machines?

Why?

- ***Adoption:***
  - Track IPv6 infrastructure evolution, how deployed
- ***Bootstrapping:***
  - IPv6 geolocation, reputation by correlating to IPv4 counterpart
- ***Security:***
  - Better understand correlated failures
  - Lack of IPv6 security, tunnel to circumvent firewalls
  - (e.g. an attack on IPv6 resource affecting IPv4 service)
- ***Performance:***
  - Isolate path vs. host performance when comparing IPv4 and IPv6

Motivation

Question?

Is IPv6 infrastructure being deployed with separate hardware or by adding IPv6 to existing machines?

Why?

- ***Adoption:***
  - Track IPv6 infrastructure evolution, how deployed
- ***Bootstrapping:***
  - IPv6 geolocation, reputation by correlating to IPv4 counterpart
- ***Security:***
  - Better understand correlated failures
  - Lack of IPv6 security, tunnel to circumvent firewalls
  - (e.g. an attack on IPv6 resource affecting IPv4 service)
- ***Performance:***
  - Isolate path vs. host performance when comparing IPv4 and IPv6

Contributions

IPv4/IPv6 Server Sibling Inference, Contributions

1. Develop an active IPv4/IPv6 sibling inference measurement technique by extending prior fingerprinting work
2. Validate and evaluate technique on ground-truth
3. Use technique to survey top Alexa IPv6 capable web servers

Outline

Sibling Identification

Targeted, Active Sibling Identification

- Intuition: IPv4 and IPv6 share a common transport-layer (TCP)
- Combine, extend, and reappraise prior TCP fingerprinting work:
  - ***Coarse-grained:*** TCP options signature [Nmap]
  - ***Fine-grained:*** TCP timestamp clockskew [Kohno 2005]

Course-Grained Sibling Identification

Course-Grained Sibling Identification

- Presence of TCP options is common-case
- Order and packing of options is implementation dependent, e.g.:
  - Win: `<mss, nop, wscale 5, nop, nop, TS, sackOK>`
  - FreeBSD: `<mss, nop, wscale 3, sackOK, TS>`
  - Linux: `<mss, sackOK, TS, nop, wscale 4>`
- We:
  - Strip timestamp value
  - Strip MSS value (unreliable, not just IPv4 MSS-20)
  - Preserve order, compare between IPv4 and IPv6

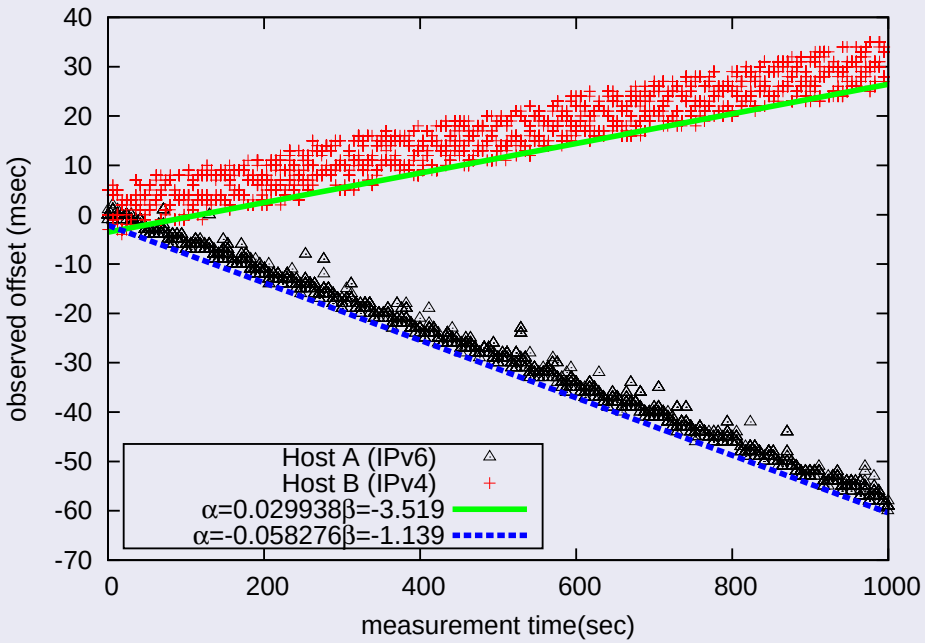
Fine-Grained Sibling Identification

Fine-Grained Sibling Identification

- TCP timestamp option: “TCP Extensions for High Performance” [RFC1323, May 1992]. Universally supported, enabled by default.
  - Option value: 4 bytes containing current clock
  - TS clock:
    - Value not specified in RFC (only used to detect duplicate segments)
    - $\neq$ system clock
    - Frequently unaffected by system clock adjustments (e.g. NTP)
- Connect to remote TCP periodically over time, fetch TS
- Fingerprint is TS clock *skew* or *drift*

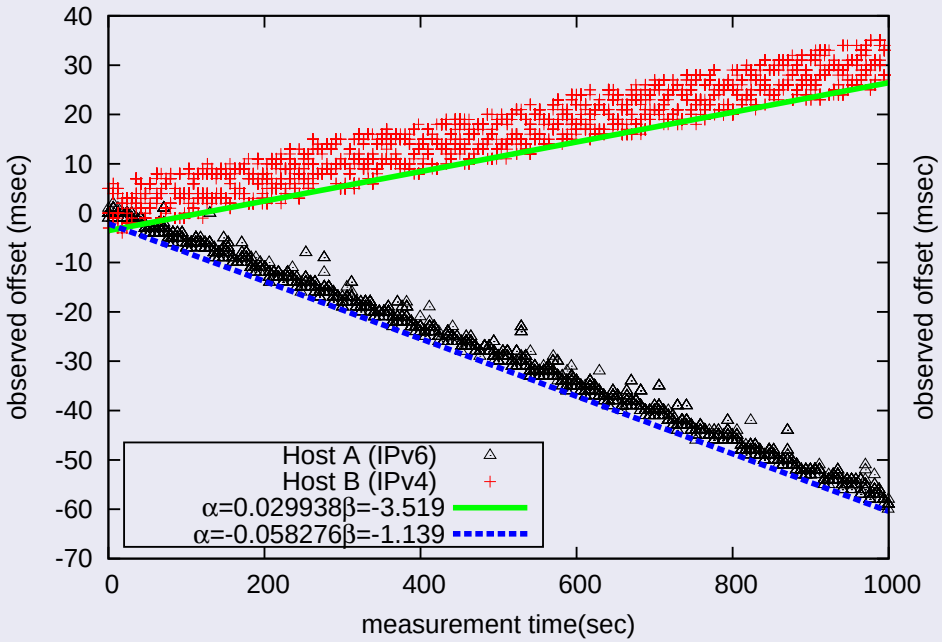
TCP Timestamp Clock Skew

Skew-based Fingerprinting Idea:

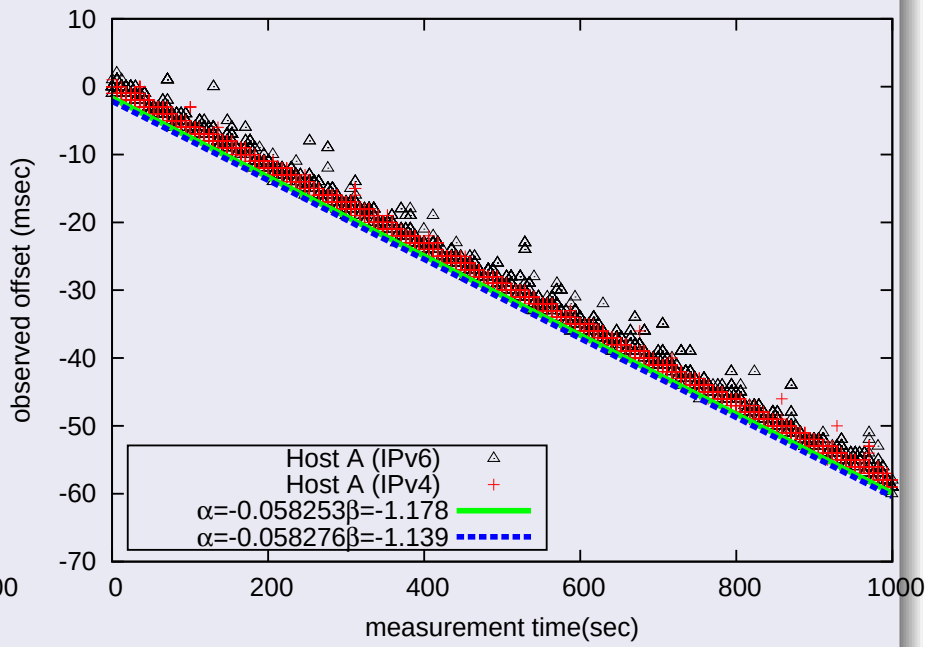


- Use linear program to find slope of points
- Here, different skews (one negative)
- $y = 0.0299x$ skew ($\approx$ 1.8ms/min, $\approx$ 15 min/year)
- Then:
  - Compare IPv4 and IPv6 slopes
  - Siblings if angle less than threshold

Example: Ground Truth Visualization



Non-Siblings



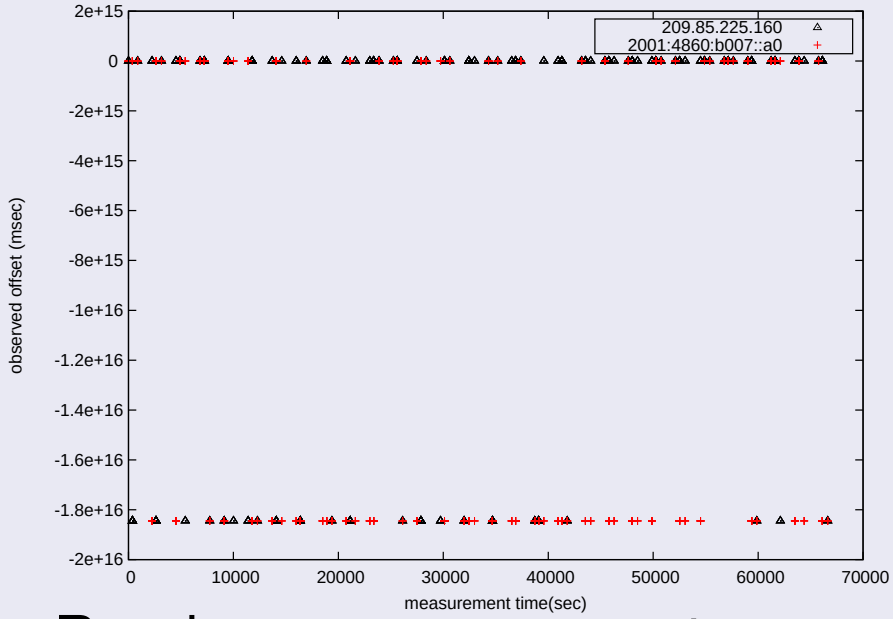
Siblings

- Host A IPv4 vs. Host A IPv6: identical slopes ($\theta = 0.0098$)
- Host A IPv6 vs. Host B IPv4: different slopes ($\theta = 31.947$)

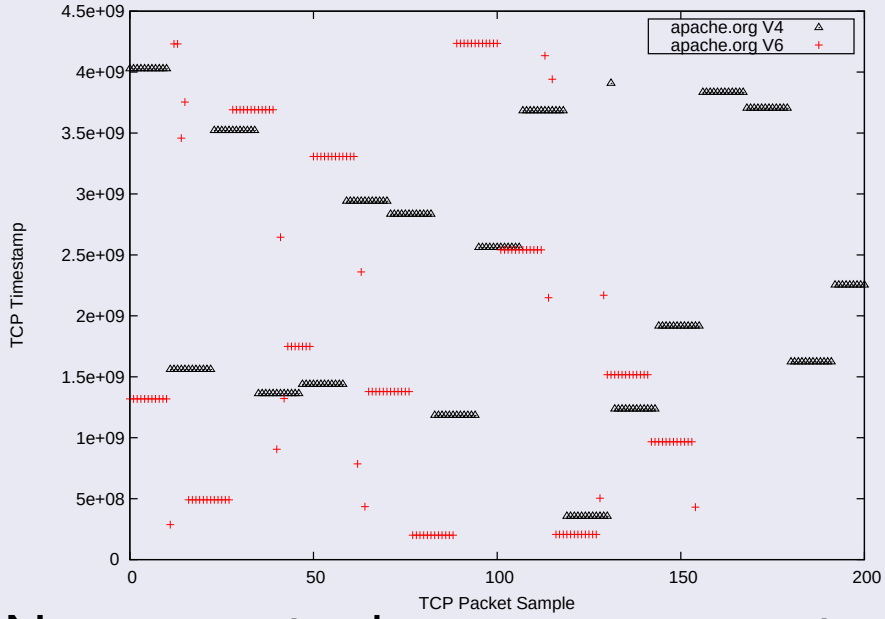
Of course, more complicated in practice!

Probing Outcomes

- ***No options returned:*** Infrequent, limits to coarse
- ***Timestamps:***
  - Not present: e.g., middlebox, limits to coarse
  - Non-monotonic: (between connections) e.g., load-balancer
  - Random: e.g., BSD's random per-flow offset
  - Monotonic: fine-grained fingerprinting
- For example, raw TCP timestamps:



Random across connects



Non-monotonic across connects

Methodology

Server Sibling Inference

- Propose and evaluate two algorithms:
  1. Options signature and basic timestamp skew (Alg 1)
  2. Additional, parameterized logic (Alg 2)
- (See paper for gory algorithm details)
- Test against ground truth
- Periodically probe Alexa IPv4 and IPv6 targets once every ∼3.5 hours for ∼17 days

Outline

Ground Truth Validation

	Hosts	# v4 AS	# v6 AS	Countries	# Option Signatures
Ground Truth	61	34	34	19	13

Ground Truth:

- Friends and family
- Small, but well-distributed: among ASes, countries, and OSes
- Permits $\sim 1,800$ combinations of non-siblings

Ground Truth Evaluation

- Ten rounds of testing, forming equal number random (known) non-siblings
- Option signatures alone: $\sim$ 82% accuracy
- Timestamps alone: $\sim$ 91% accuracy
- Combined algorithms perform best on our ground truth
- Note: high precision and specificity, but at cost of more indeterminate predictions

Validation Results

Algorithm	Acc.	Prec.	Recall	Specif.	Unknown
TCP Opts	82.2%	74.1%	98.2%	66.8%	0.0%

Ground Truth Evaluation

- Ten rounds of testing, forming equal number random (known) non-siblings
- Option signatures alone: $\sim$ 82% accuracy
- Timestamps alone: $\sim$ 91% accuracy
- Combined algorithms perform best on our ground truth
- Note: high precision and specificity, but at cost of more indeterminate predictions

Validation Results

Algorithm	Acc.	Prec.	Recall	Specif.	Unknown
TCP Opts	82.2%	74.1%	98.2%	66.8%	0.0%
Kohno	90.6%	82.3%	97.0%	86.4%	27.8%

Ground Truth Evaluation

- Ten rounds of testing, forming equal number random (known) non-siblings
- Option signatures alone: $\sim$ 82% accuracy
- Timestamps alone: $\sim$ 91% accuracy
- Combined algorithms perform best on our ground truth
- Note: high precision and specificity, but at cost of more indeterminate predictions

Validation Results

Algorithm	Acc.	Prec.	Recall	Specif.	Unknown
TCP Opts	82.2%	74.1%	98.2%	66.8%	0.0%
Kohno	90.6%	82.3%	97.0%	86.4%	27.8%
Alg 1	94.2%	93.6%	91.4%	96.0%	22.4%
Alg 1&2	97.4%	99.6%	93.1%	99.8%	29.4%

Datasets

	Hosts	# v4 AS	# v6 AS	Countries	# Option Signatures
Alexa embedded	1050	85	80	31	30
Alexa non-CDN	1533	629	575	69	73
Alexa CDN	230	59	55	18	29

Alexa:

- Top 100,000 sites with both `A` and `AAAA` records
- Remove duplicate addresses
- Subdivide into:
  - ***Embedded:*** IPv4 address encoded into IPv6 address
  - ***CDN:*** Geographically dispersed servers supporting domain
  - ***non-CDN:*** Remainder
- Well-distributed: among ASes, countries, OSes

Alexa Machine-Sibling Inferences

Inference	non-CDN	CDN	Embed
Siblings	816 (53.2%)	55 (23.9%)	978 (93.1%)
Non-Siblings	409 (26.7)	98 (42.6)	31 (3.0)
Unknown	308 (20.0)	77 (33.5)	41 (3.9)
Total	1533 (100%)	230 (100%)	1050 (100%)

- Sibling prevalence: Embedded > non-CDN > CDN

Alexa Machine-Sibling Inferences

Inference	non-CDN	CDN	Embed
Siblings	816 (53.2%)	55 (23.9%)	978 (93.1%)
Non-Siblings	409 (26.7)	98 (42.6)	31 (3.0)
Unknown	308 (20.0)	77 (33.5)	41 (3.9)
Total	1533 (100%)	230 (100%)	1050 (100%)

- Surprisingly, 3.0% of embedded are non-siblings
- Highlights that addresses alone do not imply siblings!

Alexa Machine-Sibling Inferences

Inference	non-CDN	CDN	Embed
Unknown			
- v4 and v6 missing	196 (12.8%)	6 (2.6%)	26 (2.5%)
- v4 and v6 random	32 (2.1%)	25 (10.9%)	6 (0.6%)

- Load balancers primary source of unknowns:
  - Missing timestamps for 12.8% of non-CDN
  - Operator feedback: missing timestamps due to front-end load balancer
  - Non-monotonic for 19.6% of CDN (inherent load balancing)

Alexa Machine-Sibling Inferences

Inference	non-CDN	CDN	Embed
Unknown			
- v4 and v6 missing	196 (12.8%)	6 (2.6%)	26 (2.5%)
- v4 and v6 random	32 (2.1%)	25 (10.9%)	6 (0.6%)
- v4 and v6 non-mono	78 (5.1%)	45 (19.6%)	9 (0.9%)
- v4 or v6 unresp.	2 (0.1%)	1 (0.4%)	0 (0.0%)

- Load balancers primary source of unknowns:
  - Missing timestamps for 12.8% of non-CDN
  - Operator feedback: missing timestamps due to front-end load balancer
  - Non-monotonic for 19.6% of CDN (inherent load balancing)

Autonomous System (AS) Agreement

- Examine origin AS of routeviews prefixes for addresses
- IPv4 and IPv6 addresses more likely to be in same AS when siblings
- CDN (both sibling and non-sibling) least likely to have addresses in same AS
- 10% of non-CDN and 2.7% of embedded siblings are in *different* ASes!

Sibling Inference AS Agreement

	Fraction of matching (l^4, l^6) ASNs		
Inference	**non-CDN**	**CDN**	**Embedded**
Siblings	90.0%	83.6%	97.3%
Non-Siblings	78.2%	51.0%	87.1%
Unknown	91.6%	62.3%	78.0%

Autonomous System (AS) Agreement

- Examine origin AS of routeviews prefixes for addresses
- IPv4 and IPv6 addresses more likely to be in same AS when siblings
- CDN (both sibling and non-sibling) least likely to have addresses in same AS
- 10% of non-CDN and 2.7% of embedded siblings are in *different* ASes!

Sibling Inference AS Agreement

	Fraction of matching (l^4, l^6) ASNs		
Inference	**non-CDN**	**CDN**	**Embedded**
Siblings	90.0%	83.6%	97.3%
Non-Siblings	78.2%	51.0%	87.1%
Unknown	91.6%	62.3%	78.0%

Summary

- Integration and refinement of fingerprinting methods to actively test server IPv4/IPv6 sibling relationships
- Evaluation of technique on ground-truth with >97% accuracy and 99% precision
- Survey of Alexa top 100,000 site server sibling relationships
- Even embedded IPv4 addresses do not imply IPv4/IPv6 siblings (or even same AS)

Thanks!

Questions?

`http://www.cmand.org/ipv6/`

Backup

Alexa Machine-Sibling Inferences

Inference	non-CDN	CDN	Embed
Siblings			
- v4/v6 drift match	816 (53.2%)	55 (23.9%)	978 (93.1%)
Non-Siblings			
- v4 and v6 opt sig diff	229 (14.9%)	14 (6.1%)	22 (2.1%)
- v4 or v6 missing	70 (4.6%)	11 (4.8%)	7 (0.7%)
- v4 or v6 random	23 (1.5%)	13 (5.7%)	1 (0.1%)
- v4 or v6 non-mono	52 (3.4%)	47 (20.4%)	1 (0.1%)
- v4/v6 drift mismatch	35 (2.3%)	13 (5.7%)	0 (0.0%)
Unknown			
- v4 and v6 missing	196 (12.8%)	6 (2.6%)	26 (2.5%)
- v4 and v6 random	32 (2.1%)	25 (10.9%)	6 (0.6%)
- v4 and v6 non-mono	78 (5.1%)	45 (19.6%)	9 (0.9%)
- v4 or v6 unresp.	2 (0.1%)	1 (0.4%)	0 (0.0%)
Total	1533 (100%)	230 (100%)	1050 (100%)