Active Server Sibling Resolution

Robert Beverly, Arthur Berger*

Naval Postgraduate School
*MIT/Akamai
rbeverly@nps.edu, awberger@mit.edu

January 7, 2013

NPS IPv6 Measurement Meeting 2013

Outline

Sibling Resolution

New Problem We Term "Sibling Resolution:"

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

- Sibling resolution may be either active or passive.
- Lots of prior work on passive sibling associations: e.g. web-bugs, javascript, etc.
- Prior work focuses on clients (adoption, performance)
- This work:
  - *Targeted, active test:* on-demand for any given pair
  - *Infrastructure:* finding server siblings

Motivation

Why?

- IPv4 and IPv6 expected to co-exist (for a long while?) $\rightarrow$ dual-stacked devices
  - Track adoption (and dis-adoption)
  - Track IPv6 evolution
- Security:
  - Inter-dependence of IPv6 on IPv4 (and vice-versa)
  - e.g. attack on IPv6 resource affecting IPv4 service
- Performance:
  - Measurements of IPv4 vs. IPv6 performance
  - Desire to isolate path vs. host performance
  - Correlating geolocation, reputation, etc with IPv4 host counterpart.

Outline

Targeted, Active Technique

Targeted, Active Technique

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

TCP Timestamp Clock Skew

TCP Timestamp Clock Skew

- TS value: 4 bytes containing current clock
- Note: RFC does not specify value of TS (assume millisec for now)
- Note: TS clock $\neq$ system clock
- Note: TS clock frequently unaffected by system clock adjustments (e.g. NTP)
- ***Basic Idea:*** Probe over time. Fingerprint is clock *skew* (and remote clock resolution).

TCP Timestamp Clock Skew

Some Details

- Must be able to connect to remote TCP service on each host
- Periodically connect to TCP service.
- Given a sequence of timestamp offsets, use linear programming to obtain a line that minimizes distance to points, constrained to be under data points.
- Obtain: $y_4 = \alpha_4 x + \beta_4$ and $y_6 = \alpha_6 x + \beta_6$
- Angle between lines then:

$$\theta(\alpha_4, \alpha_6) = \tan^{-1} \left| \frac{\alpha_4 - \alpha_6}{1 + \alpha_4 \alpha_6} \right|$$

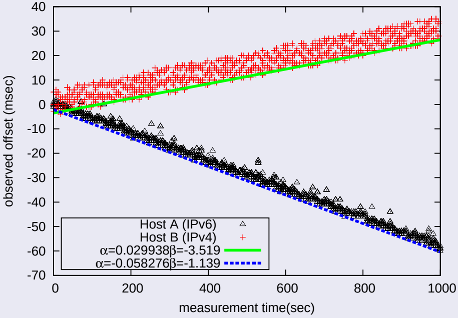
- Siblings if: $\theta < \tau$

Example

Example

- Gather 4 timestamp series:
  - `www.caida.org` (v4 and v6)
  - `www.ripe.net` (v4 and v6)

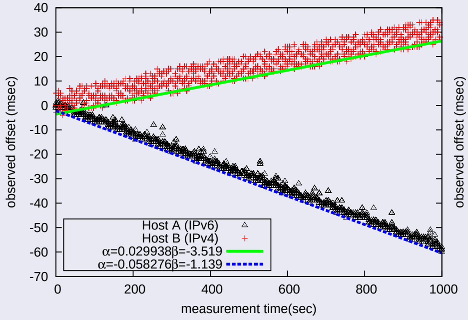
Example



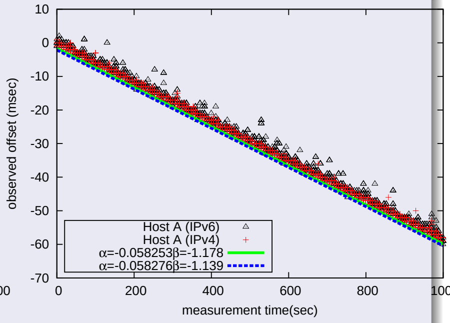
CAIDA IPv6 vs. RIPE IPv4

- Observe different skew slopes (one negative)
- Different timestamp granularity
- $y = 0.029938x$ equates to skew of ≈ 1.8ms / minute, or ≈ 15 minutes per year.
- False siblings!

Example



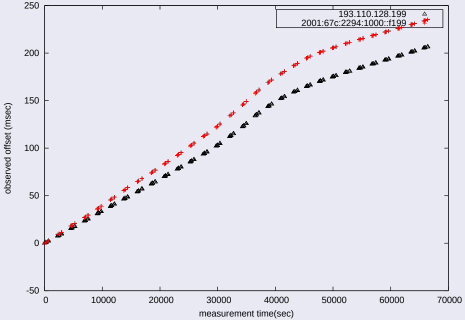
False Siblings



True Siblings

- CAIDA IPv4 vs. CAIDA IPv6: identical slopes ($\theta = 0.0098$)
- CAIDA IPv6 vs. RIPE IPv4: different slopes ($\theta = 31.947$)

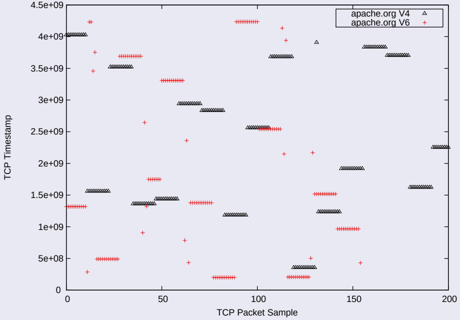
Complications



www.marca.com (#6 on alexa ipv6)

- Not always so distinct of a difference!
- Slope angle difference: $\theta = 2.046$

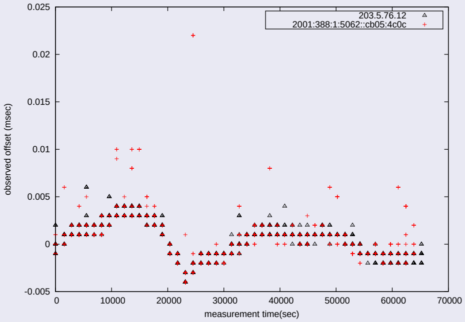
Complications



www.apache.com

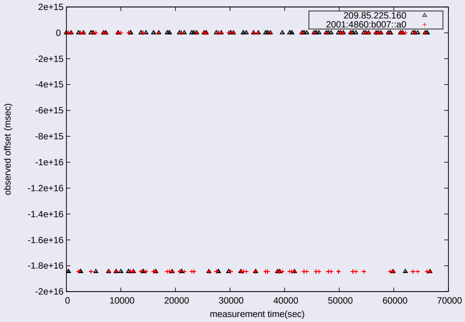
- Raw TCP timestamps
- Deterministically random and monotonic for a single connection
- Random across connections. Looks like noise to us.

Complications



- What's going on here?

Complications



- Also detects load balancing among servers
- But how to deal with it?

Outline

Machine Sibling Inference

Machine Sibling Inference Methodology:

- Analyze Alexa top 100,000 websites
- Pull `A` and `AAAA` records
- 1398 ($\approx$ 1.4%) have IPv6 DNS
- Repeatedly fetch root HTML page via IPv4 and IPv6 via deterministic IP address
- Record all packets

Machine Sibling Inference

Alexa 100K Targeted Machine-Sibling Inference

Case	Count
v4 and v6 non-monotonic (possible siblings)	109 (7.8%)
v4 or v6 non-monotonic (non-siblings)	140 (10.0%)
v4 and v6 no timestamps (possible siblings)	94 (6.7%)
v4 or v6 no timestamps (non-sibling)	101 (7.2%)

- 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)
- Note, can disambiguate non-siblings

Machine Sibling Inference

Alexa 100K Targeted Machine-Sibling Inference

Case	Count
v4 and v6 non-monotonic (possible siblings)	109 (7.8%)
v4 or v6 non-monotonic (non-siblings)	140 (10.0%)
v4 and v6 no timestamps (possible siblings)	94 (6.7%)
v4 or v6 no timestamps (non-sibling)	101 (7.2%)
Skew-based siblings	839 (60.0%)
Skew-based non-siblings	115 (8.3%)
Total	1398 (100%)

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

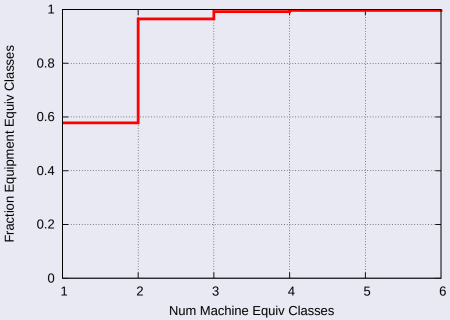
DNS Machine Siblings

DNS Machine Siblings

- With respect to collecting DNS siblings, would like to differentiate between *machine* and *equipment* siblings.
- Tie passive and active DNS collection with skew-based inference.
- For addresses with an DNS equivalence class:
  - Add IP to machine sibling group with small $\theta < 1.0$
  - Else $\theta \geq 1.0$, create new sibling group with single IP.
  - Until all IPs of equipment equivalence class clustered

DNS Machine Siblings

DNS Machine Siblings



Relationship between equipment siblings and machine siblings.

Evaluating Sibling Inference Accuracy

Evaluating Inference Accuracy

- Seek to understand the accuracy of timestamp-based sibling inference
- Use ground-truth dual-stacked Akamai machines
- No load-balancers or middleboxes
- Experiment: 100 known-siblings, 100 known non-siblings (random v4/v6 pairs drawn from Akamai population)
- *Hardest scenario*: single organization, similar boxes, same operating system, etc.

Evaluating Sibling Inference Accuracy

Evaluating Inference Accuracy

		Prediction	
		sibling	**non**
Actual	**sibling′**	84 TP	13 FN
	non′	43 FP	54 TN

- Threshold $\tau = 0.002$ gives best results!
- 71% accuracy, 66% precision, 87% recall (f-score: 0.75)

Evaluating Sibling Inference Accuracy

Evaluating Inference Accuracy

Prediction

		sibling	**non**
Actual	**sibling′**	97 TP	0 FN
	non′	94 FP	3 TN

- No false negatives w/ $\tau = 0.05$ (but more FP's)
- 52% accuracy, 51% precision, 100% recall (f-score: 0.67)

Current Work

Current Work

- Quantify whether vantage point imparts any difference on results
- Refine inference algorithm to deal with load-balancers
- Refine algorithm to produce better accuracy, eliminate false positives