High-Frequency Active Internet Topology Mapping

Robert Beverly[*], Geoffrey Xie[*], Ralucca Gera[◇], Justin Rohrer[*], Arthur Berger[†], Guillermo Baltra[*], Erik Rye[*], Jamar Wright[◇]

Naval Postgraduate School
[*] Computer Science Dept.
[◇] Applied Math Dept.
[†] Akamai Technologies

March 14, 2014

DHS BAA11-02 PI Meeting

Outline

1. Project Overview
2. Background
3. Current Implementation
4. Other Topology Work

Project Overview

High-Frequency Active Internet Topology Mapping:

- DHS S&T BAA-11-02 Cyber Security Division
- TTA #7 “Network Mapping and Measurement”
- Q4 2012 – Q4 2015

This presentation covers our midway ($\sim$ 1.5 year) project progress

Project Objective

Goal:

- Obtain accurate network graphs, at interface and router granularities, even at large scale (e.g. Internet) and amid topological sparsity (e.g. IPv6).
- Obtain topologies an order of magnitude faster than existing systems in order to better capture transient dynamics, including malicious or misconfiguration events.
- Working systems implementation, with transfer to production mapping systems.

Project Objective

Goal:

- Obtain accurate network graphs, at interface and router granularities, even at large scale (e.g. Internet) and amid topological sparsity (e.g. IPv6).
- Obtain topologies an order of magnitude faster than existing systems in order to better capture transient dynamics, including malicious or misconfiguration events.
- Working systems implementation, with transfer to production mapping systems.

Project Objective

Goal:

- Obtain accurate network graphs, at interface and router granularities, even at large scale (e.g. Internet) and amid topological sparsity (e.g. IPv6).
- Obtain topologies an order of magnitude faster than existing systems in order to better capture transient dynamics, including malicious or misconfiguration events.
- Working systems implementation, with transfer to production mapping systems.

Why care? : DHS BAA 2011-02:

"The protection of cyber infrastructure depends on the ability to identify critical Internet resources, incorporating an understanding of geographic and topological mapping of Internet hosts and routers. A better understanding of connectivity richness among ISPs will help to identify critical infrastructure. Associated data analysis will allow better understanding of peering relationships, and will help identify infrastructure components in greatest need of protection. Improved router level maps (both logical and physical) will enhance Internet monitoring and modeling capabilities to identify threats and predict the cascading impacts of various damage scenarios."

These proposed capabilities are critical to U.S. national security missions, analyses of cyber infrastructure threats and risks, and hardening of U.S. military, as well as civilian, Internet communications environments.

Network Mapping

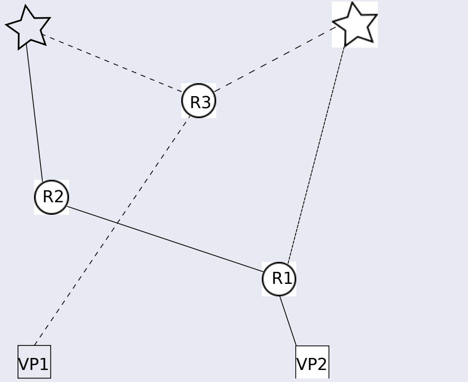
Motivation:

- Protect and improve critical infrastructure
- Understand structural properties of the Internet topology, including robustness, vulnerability to attack, potential for correlated failures, IPv4/IPv6 interdependence, etc.
- Enabler of other work
  - Understanding peering/interconnection
  - Data vs. control plane correlation
  - Evolution/longitudinal studies
  - CDN optimization

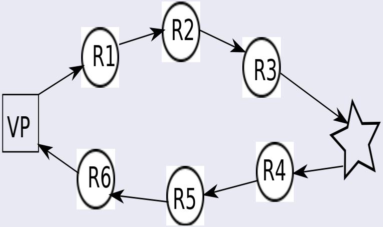
Network Mapping Examples

- Enabler of other security work:

Geolocation:



Reverse Traceroute:



The Problem

Why it's hard:

- A large, complex distributed system (organism)
- Non-stationary (in time)
- Difficult to observe, multi-party (information hiding)
- Poorly instrumented (not part of original design)
- Lack of ground truth

⇒ Measurement community making continued progress in understanding network topology (interface, router, AS, or organization level)

The Problem

State-of-the-Art

- Significant prior work, but not a solved problem
- Production topology mapping systems (e.g. iPlane, Ark) must balance measurement load vs. fidelity
- Takes several days to obtain an (incomplete) network map
- Mapping time especially important for alias resolution, IPv6, etc.
- Can miss transient dynamics (e.g. Nyquist sampling loss), which might reveal properties of interest
- Our project seeks to advance state-of-the-art and to be complimentary to existing work

Status Highlights

Progress on Deliverables:

1. Updated our originally proposed topology primitives after real-world experience
2. Implementation of Recursive Subnet Inference (RSI) and Ingress Point Spreading (IPS) algorithms on CAIDA's Ark infrastructure
3. Integration of RSI and IPS, and operational experience using our algorithms
4. Demonstrate discovering *more* topology with *fewer* probes
5. Close working relationship with CAIDA and Akamai; alpha quality code shared with CAIDA

Status Highlights

Topology Publications:

1. Baltra, Beverly, Xie, "*Ingress Point Spreading: A New Primitive for Adaptive Active Network Mapping*," in Passive and Active Measurement (PAM) Conference, Mar, 2014.
2. Luckie, Beverly, Claffy, "*Speedtrap: Internet-Scale IPv6 Alias Resolution*," in Internet Measurement Conference (IMC), Nov, 2013.
3. Berger, Weaver, Beverly, Campbell, "*Internet Nameserver IPv4 and IPv6 Address Relationships*," in Internet Measurement Conference (IMC), Nov, 2013.

Talk Outline

Items for PI Meeting:

- Background and Prior Work
- Current Implementation Status
- Future Work
- Other Topology Work

Outline

1 Project Overview

2 Background

3 Current Implementation

4 Other Topology Work

Our Prior Work

"*Primitives for Active Internet Topology Mapping: Toward High-Frequency Characterization*", IMC 2010.

- Investigate current production topology mapping systems
- Ark/Skitter (CAIDA), iPlane (UW)
- *Multiple days and significant resources for complete cycle*

Adaptive Probing Methodology

We develop three primitives:

1. Subnet Centric Probing
2. Vantage Point Spreading
3. Interface Set Cover

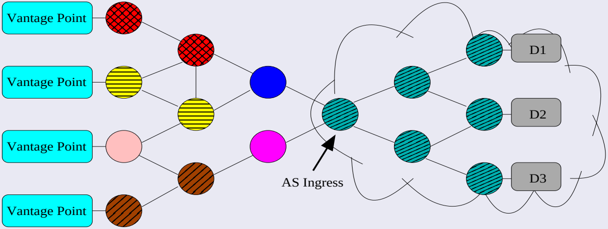
These primitives leverage adaptive sampling, external knowledge (e.g., common subnetting structure, BGP, etc), and data from prior cycles to *maximize efficiency and information gain of each probe*.

Adaptive Probing Methodology

We develop three primitives:

1. Subnet Centric Probing
2. Vantage Point Spreading
3. Interface Set Cover

Best explained by understanding sources of path diversity:

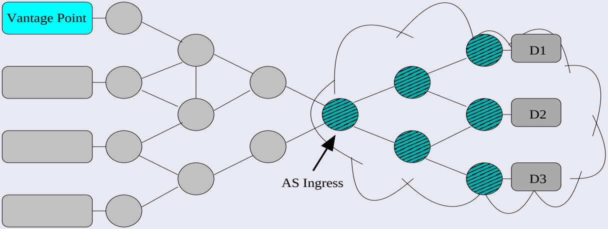


Subnet Centric Probing

Granularity vs. Scaling

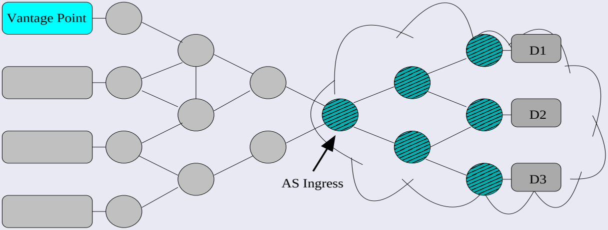
- $\sim 2^{32-1}$ possible destinations (~2.9B in routeviews)
- What granularity? /24's? Prefixes? AS's?

Subnet Centric Probing



- From a single vantage point, no path diversity into the AS
- Path diversity due to AS-internal structure

Subnet Centric Probing



- **Goal:** adapt granularity, discover internal subnetting structure
- Leverage BGP as coarse structure
- Follow *least common prefix:* iteratively pick destinations within prefix that are maximally distant (in subnetting sense)
- Address "distance" is misleading: e.g. `18.255.255.100` vs. `19.0.0.4` vs. `18.0.0.5`

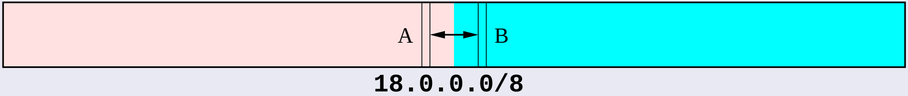
Adaptive Sampling

Least-Common Prefix:

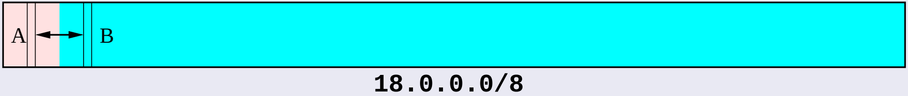
- Use knowledge of how networks are provisioned and subnetted

Penalizing Complexity:

Easier to believe *A* and *B* in different subnets:

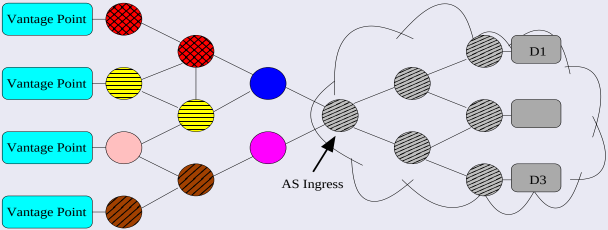


than *A*′ and *B*′ in different subnets:



Vantage Point Spreading

Vantage Point Spreading

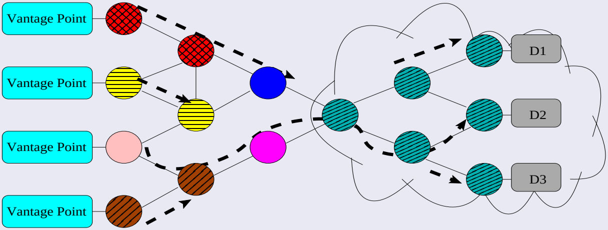


- Discover AS ingress points and paths to the AS via multiple vantage points
- Random assignment of destinations to vantage points is wasteful

Interface Set Cover

Interface Set Cover

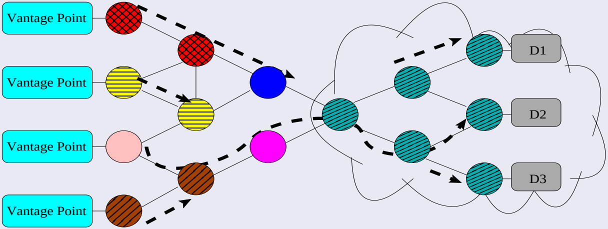
- As shown in preceding analysis, full traces very inefficient
- Perform greedy minimum set cover approximation (NP-complete)
- Select subset of *prior* round probe packets for *current* round



Interface Set Cover

Interface Set Cover

- Generalizes DoubleTree [DRFC05] without parametrization
- Efficient
- Inherently multi-round



Our Prior Work

"*Primitives for Active Internet Topology Mapping: Toward High-Frequency Characterization*", IMC 2010.

- Demonstrated the ability of each primitive to generate significant probing savings. Fewer probes implies potential to:
  - Improve quality of topologies as currently inferred
  - Perform additional probing for e.g. alias resolution using same probing "budget"
  - Perform more complete/detailed probing
  - Increase feasible frequency (i.e. speed) of full-topology inferences

Our Prior Work

That was then, this is now

- Limitations:
  - Primitives examined *in isolation*
  - Performance of primitives simulated by selectively using/ignoring probes in CAIDA traces
- Project Deliverables:
  - Real-world implementation of three primitives on CAIDA's Ark platform
  - Integration of three primitives
  - Analysis of performance
  - Technology transfer: integration into Ark

Outline

1 Project Overview

2 Background

3 Current Implementation

4 Other Topology Work

From Theory to Practice

Project Deliverable:

- Implement SCP, ISC, VPS on CAIDA’s Ark
- Ark provides a straight-forward API for performing asynchronous traces easily – abstracts work of communicating with distributed collection of vantage points.
- Easy, right?

Improving Ark Interface/API

Ark API

- Asynchronous “tuple space” abstracts much of the measurement complexity
- Which is great until you can’t figure out what’s going on

Contribution 1:

- `youngh@caida`: “First group to really stress Ark API”
- Worked with CAIDA to identify and fix Ark bugs with probe request queue getting stuck
- Led to per-session multiplexing for concurrent measurement sessions
- Led to `tod-debug` which provides user introspection into request and response tuple spaces, and ability to clear queues

Naïve SCP

Naïve SCP

- Our simulation using CAIDA traces used edit distance (ED) as stopping criterion on recursion.

Load balancing and Edit Distance:

- Artificially distorts ED for some paths
- Examining traces in a purely pair-wise fashion (without regard for prior traces to *same* prefix) is short-sighted
- In practice: recurse all the way down to /32s. ☹
- Note: this occurs even when using Paris-style traceroute. Paris ensures determinism over per-flow load balanced path to a given destination. SCP uses *different* destinations as part of its exploration algorithm.

Probing Strategy

Recursive Subnet Inference (RSI)

- Designed to discover the degree of subnetting within networks through an iterative interrogation process.
- Performs a binary search over the target network's address space pruning those branches of the tree that do not reveal new topology information.
- RSI receives as input a network prefix. The address space is divided into 2 halves and probes the center address of each half as defined by the LCP algorithm.
- If a returning probe provides newly discovered interfaces, the procedure is repeated by dividing the corresponding address space into smaller subparts.

Improved SCP

RSI, Key Ideas:

1. ***Focus on destination AS:*** RSI's objective is to discover structure *within the destination AS*. RSI should base its operation on new structure (edges, vertices) discovered in target AS.
2. ***Integrate Vantage Point Spreading:*** By focusing on target AS, we can distribute the source of probes. Using multiple vantage points as part of RSI naturally helps discover AS ingress points.
3. ***Maintain State:*** RSI's recursive stopping criterion should consider all traces to a destination prefix, rather than just being pair-wise.

RSI

Input:

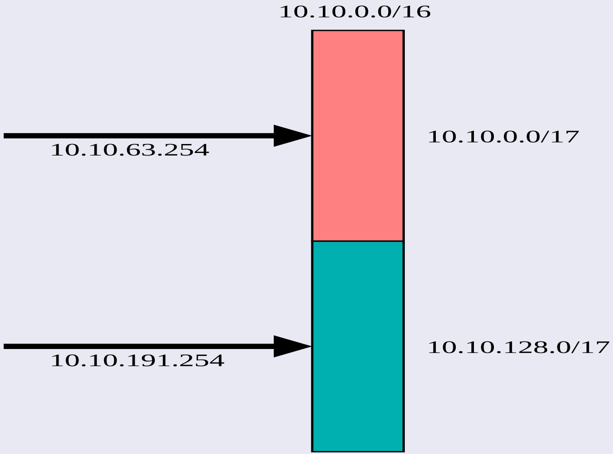
p/m:	Destination prefix / mask
M:	Set of (rank-ordered) monitors
τ:	Threshold

Output:

T: Set of path traces

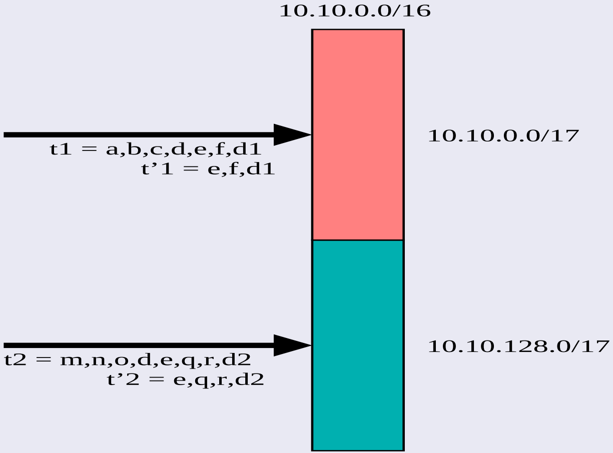
RSI, Step 1

RSI, Step 1



RSI, Step 1

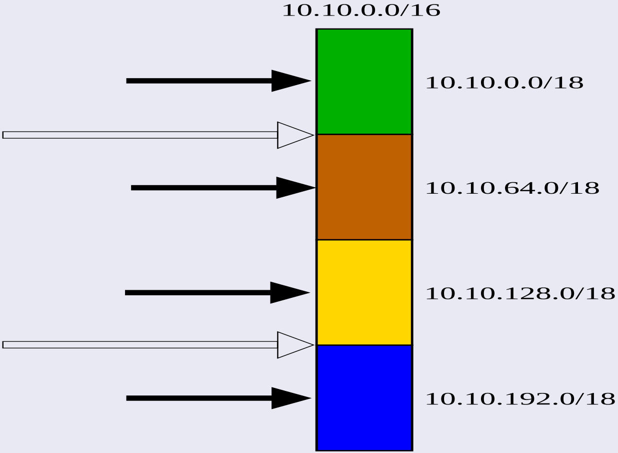
RSI, Step 1



$|t'_1 - t'_2| + |t'_2 - t'_1| > \tau$: recurse on children sub-prefixes

RSI, Step 2

RSI, Step 2



Real-World Problems

Real-World Problems

- ***No return:*** Request a trace from Ark, no response received within $\delta = 1$min. Don't block, want to make as much progress as possible (especially if probe request never return (monitor went down mid experiment)). Resend trace to same destination with a new random monitor.
- ***No comparable :*** When no interfaces along path belong to destination AS. Can cause SCP to prematurely skip a prefix. Instead, we rely on notion of "hovering:" $d' = i(-1)^i + d$

Increasing Probing Efficiency

Vantage Point Importance

- VPs used in active probing strongly influence the inferred topology (Shavitt, Weinsberg).
- Example 1:
  - CAIDA Ark system, divides the entire routed address space into logical /24 subnetworks.
  - Probes a random address within each /24 using a random VP.
  - Probing every /24 prefix once, constitutes a “cycle.”
  - Assimilates 21 cycles of probing to obtain a high resolution map.

Increasing Probing Efficiency

Vantage Point Importance

- For N cycles and M VPs, the expected number of unique VPs that explore a given /24 prefix (Y) in Ark is given by:

$$E[Y] = M - \frac{(M-1)^N}{M^{N-1}} \tag{1}$$

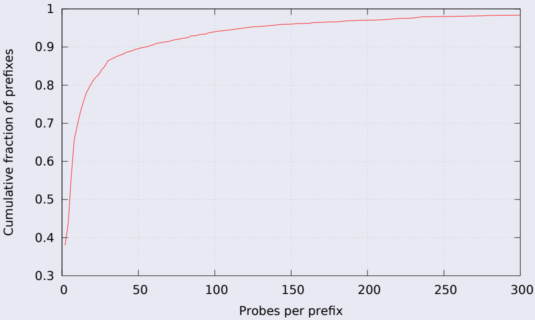
Examining one team of CAIDA probing (June, 2013) $M = 18$ VPs:

- On average, each /24 in the union of $N = 21$ cycles is explored by $E[Y] = 12.6$ VPs.

Increasing probing efficiency

Vantage Point Importance

- Example 2: RSI with 60 randomly assigned VPs probing 1500 prefixes selected at random from the global Routeviews BGP tables.

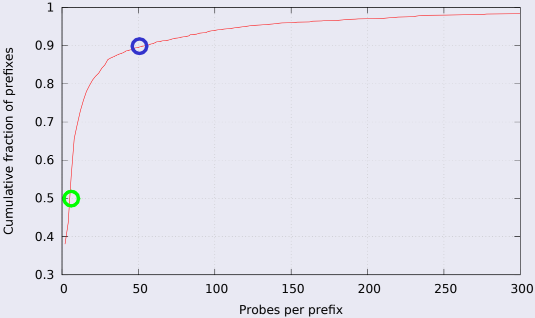


More than half of the prefixes are probed fewer than 10 times, while ∼ 90% of the prefixes see 50 or fewer probes.

Increasing probing efficiency

Vantage Point Importance

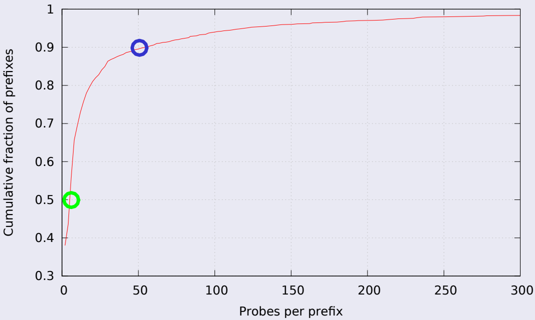
- Example 2: RSI with 60 randomly assigned VPs probing 1500 prefixes selected at random from the global Routeviews BGP tables.



More than half of the prefixes are probed fewer than 10 times, while ∼ 90% of the prefixes see 50 or fewer probes.

Increasing probing efficiency

Vantage Point Importance



More than half of the prefixes are probed fewer than 10 times, while ∼ 90% of the prefixes see 50 or fewer probes.

- The number of VPs used is frequently less than the total available.
- Even when the number of probes is larger than the number of VPs, using randomly selected VPs is sub-optimal (example 1).

VP Ordering

Does VP Ordering Matter, if we use all VPs?

- The number of VPs used for a given target network is frequently less than the total available (e.g. using RSI).
- Or, there may be a large number of VPs available
- *Therefore, the order in which VPs are employed matters.*

Improved VPS

Improved VPS; Key Ideas:

1. ***Rank-Order Monitors:*** RSI needs a “pool” of monitors when probing. Different monitors may provide different value for different target prefixes (especially with respect to discovering network ingress points, and not prematurely stopping (τ)).
2. ***Pre-probing:*** Explore value in “pre-probing” to develop a map of monitor distances.
3. ***Ingresses:*** Want to traverse all (known) ingresses into the destination network to exercise all paths

VPS^{++}

VPS^{++} Pre-Probing:

- Examined granularity
- Examined different “distance” metrics (hops, hop difference, AS difference, etc)
- Balance amount of pre-probing to get very coarse-grained structure with cost of pre-probing.

Increasing probing efficiency

Ingress Point Spreading (IPS)

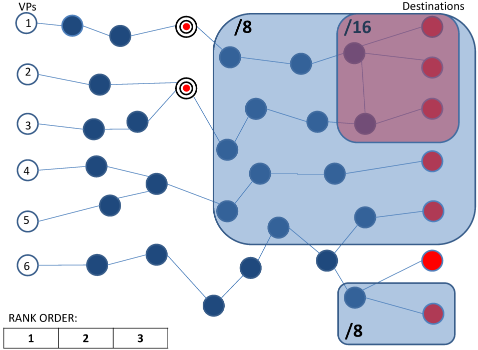
- VP selection technique, aimed to discover sources of path diversity into networks.
- Autonomous System (AS) is typically multi-homed and connected with multiple networks.
- IPS infers the number of ingress points for a given network and, then for each new probe, selects the VP with the highest likelihood to traverse a unique ingress point.
- IPS algorithm computes a per-destination network rank-ordered list of VPs based on prior rounds of probing.

Ingress Point Spreading

Notional Prefix

- An expansion to a larger prefix aggregate containing the target prefix.
- By expanding the size of the notional prefix, all VPs can be rank-ordered in order to ensure path diversity.
- *Notional prefix ingress* is the first router interface hop that leads to a next hop whose IP is within the notional prefix.
- Note: *Notional prefix* does not imply relationship to real-world BGP route aggregation.

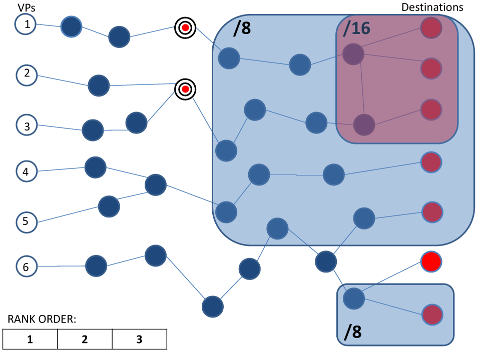
Ingress Point Spreading



e.g.

- 205.155.0.0/16 is the target prefix (red box).
- /8 is a notional prefix (blue box).
- 6 VPs used.
- Blue circles are hops.
- Red circles are destinations.
- Bullseyes are notional ingress routers.

Ingress Point Spreading

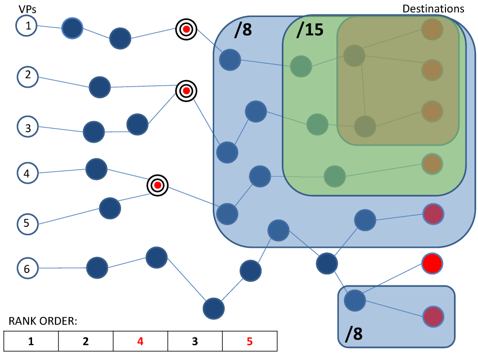


1	2	3

e.g.

- VPs 1 and 2 are selected as the first two VPs in the rank order list, (different ingresses into notional /8 prefix).
- Since VPs 2 and 3 share the same ingress router, the latter is included at the end of the list.
- However, we wish to obtain a total order over all of the VPs.

Ingress Point Spreading



RANK ORDER:

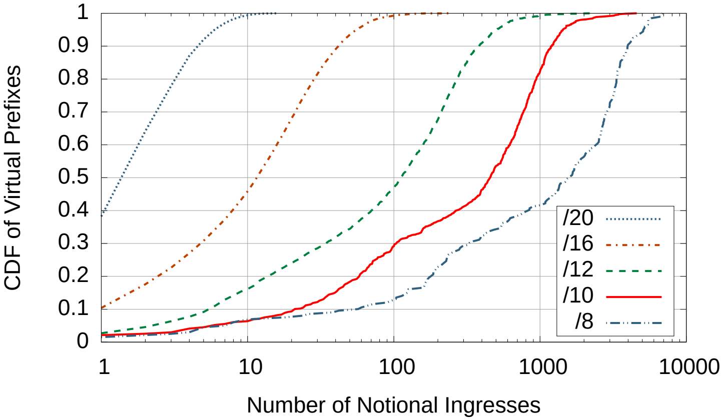
1	2	4	3	5

e.g.

- Ingress search space expansion to include `205.154.0.0/15` (green box).
- VP `4` becomes the third in the rank-order and VP `5` is included at the end of the list.
- Expansion continues until all VPs are ordered.
- i.e. `205.152.0.0/14`, `205.152.0.0/13`, ..., `205.0.0.0/8`.

Notional Prefix

Figure: Distribution of Ingresses into Prefixes of Different Logical Size



Data from CAIDA's Ark, June 2-4, 2013.

Probing Strategy

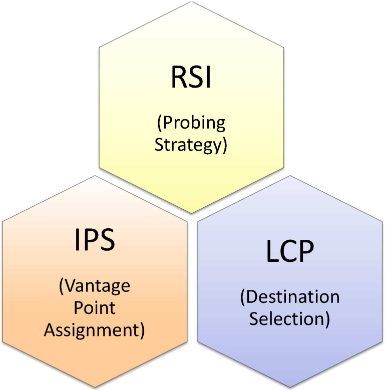


Figure: Three Step Strategy

- LCP: Least Common Prefix (Beverly, Berger, Xie [2010])
- RSI: Recursive Subnet Inference
- IPS: Ingress Point Spreading

Strategy Evaluation

IPS compared to popular mapping system, such as Ark:

- Direct comparison with published Ark data is not possible as IPS does not use “teams” of VPs.
- Emulate Ark’s methodology using the same number of VPs for both strategies.
- Pre-probing process: provide IPS with one day’s worth of CAIDA’s topology data (Aug 28, 2013), which demonstrates that IPS is not limited to our own pre-probed data.
- Using IPS and Ark’s strategy, $\sim 49k$ randomly selected prefixes were probed from 59 globally distributed VPs.

Strategy Evaluation

Metric	**Ark**	**IPS**	**IPS**
		(Aug. 2013 trained)	(Dec. 2013 trained)
Prefixes Probed	48,905	48,905	48,905
Vertices	464,544	521,513	520,903
Edges	906,680	1,024,295	1,034,101
Probes	4,041,289	2,056,562	2,052,842
Vertices (inside dest)	121,137	135,209	134,575
Vertices (intersection w/ ark)		309,997	309,971
Ingresses	31,138	38,532	39,020
Time	26h 55m	13h 38m	14h 47m

IPS is significantly more efficient:

- Using $\sim$ 50% the number of probes.
- Taking approximately half the time.
- IPS discovers 211,516 vertices not in Ark.
- Ark discovers 154,547 vertices that IPS does not.

Strategy Evaluation

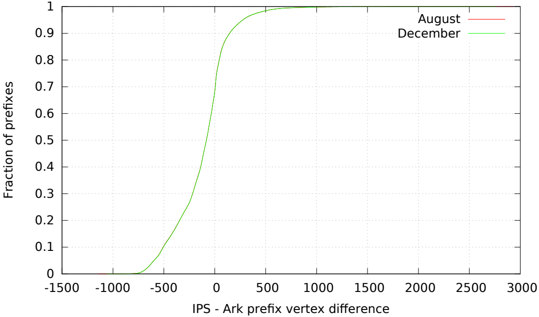
Metric	Ark	IPS (Aug. 2013 trained)	IPS (Dec. 2013 trained)
Prefixes Probed	48,905	48,905	48,905
Vertices	464,544	521,513	520,903
Edges	906,680	1,024,295	1,034,101
Probes	4,041,289	2,056,562	2,052,842
Vertices (inside dest)	121,137	135,209	134,575
Vertices (intersection w/ ark)		309,997	309,971
Ingresses	31,138	38,532	39,020
Time	26h 55m	13h 38m	14h 47m

In terms of performance of IPS against Ark:

- Top 3 prefixes are national ISP networks with hundreds of peering links.
- Bottom 3 prefixes belong to enterprise networks that have small number of peering links.

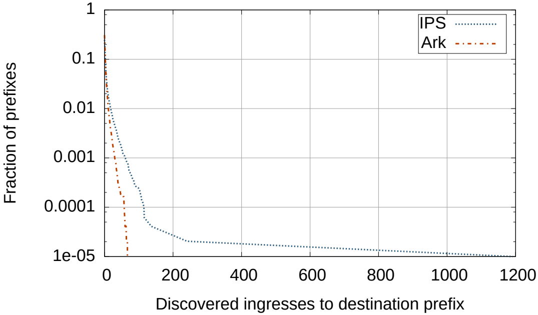
Vertex Difference

CDF of per-prefix coverage difference: *IPS – Ark*



- IPS performs worse than Ark for ∼ 66% of the prefixes.
- IPS is significantly superior to Ark for a small number of prefixes, thereby contributing to the overall superior topological coverage.

Ingress Discovery



- Among destinations where probing within the target network is feasible, IPS finds significantly more ingresses than Ark.
- Neither Ark nor IPS discovers any ingresses for $\sim$ 70% of the prefixes (ICMP blocking and other forms of packet filtering).

Future Work

While we have demonstrated promising results by utilizing ingresses to our advantage, significant future work remains:

- Scale probing by one more order of magnitude to encompass all advertised prefixes on the Internet, and run continually.
- Practical experience has shown that VPs are unreliable, yet IPS cannot simply use the next VP in the ordered list when the preferred VP is down, as the complete ordering is perturbed.
- Some prefixes with significant topology have gone undiscovered by RSI due to the particular deterministic selection of destinations causing early termination.

Outline

1 Project Overview

2 Background

3 Current Implementation

4 Other Topology Work

IPv6 Topology

Current Weakness

- IPv6 is the next generation of the Internet Protocol. (Yes, it's being adopted rapidly – economics have changed!)
- As poorly as we understand the IPv4 topology, IPv6 topology is even less well understood

IPv6 Topology

IPv6 Alias Resolution

- Luckie, Beverly, Claffy, “*Speedtrap: Internet-Scale IPv6 Alias Resolution*,” in Internet Measurement Conference (IMC), Nov, 2013.
- Too-Big Trick (TBT), implemented by CAIDA in `scamper`
- Induce a remote IPv6 router to originate fragmented packets
- Among $\approx$ 50,000 distinct IPv6 router interfaces in 2,617 ASes, works $\approx \frac{1}{2}$ the time.

Current Work:

- Characterize IPv6 router stability, up-time, etc
- Correlate with time-of-day, observed IPv6 announcements/ withdrawals in global BGP, observed IPv4 BGP behavior, etc.

Sibling Resolution

New Problem We Term “v6 Sibling Resolution:”

Given a candidate (*IPv*4, *IPv*6) address pair, determine if these addresses are assigned to the same cluster, device, or interface.

Motivation

Why?

- Adoption (non-adoption):
  - IPv4 and IPv6 expected to co-exist (for a long while?) → dual-stacked devices
  - Track IPv6 evolution
- Security:
  - IPv6 is largely unsecured!
  - Inter-dependence of IPv6 on IPv4 (and vice-versa)
  - e.g. attack on IPv6 resource affecting IPv4 service
  - Correlating geolocation, reputation, etc with IPv4 host counterpart.
- Performance:
  - Getting measurements of IPv4 vs. IPv6 performance correct: isolate path vs. host performance

Progress

Progress:

- Berger, Weaver, Beverly, Campbell, "*Internet Nameserver IPv4 and IPv6 Address Relationships*," in Internet Measurement Conference (IMC), Nov, 2013.
- Operationally deployed today in Akamai, informing Edgescape geolocation.

Sibling Resolution

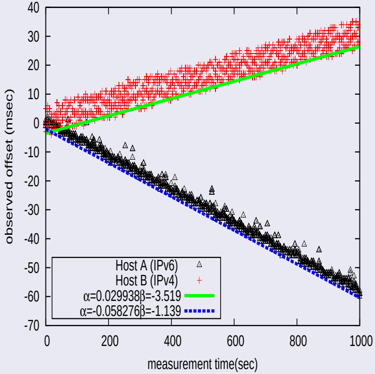
Active vs. Passive

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

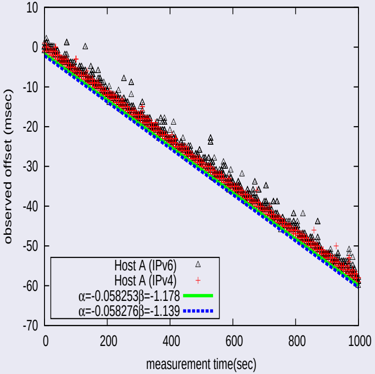
Targeted, Active Technique

- 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]. Universally supported, enabled by default.
- 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).

Sibling Resolution Example



Host A IPv6 vs. Host B IPv4 (non-siblings)



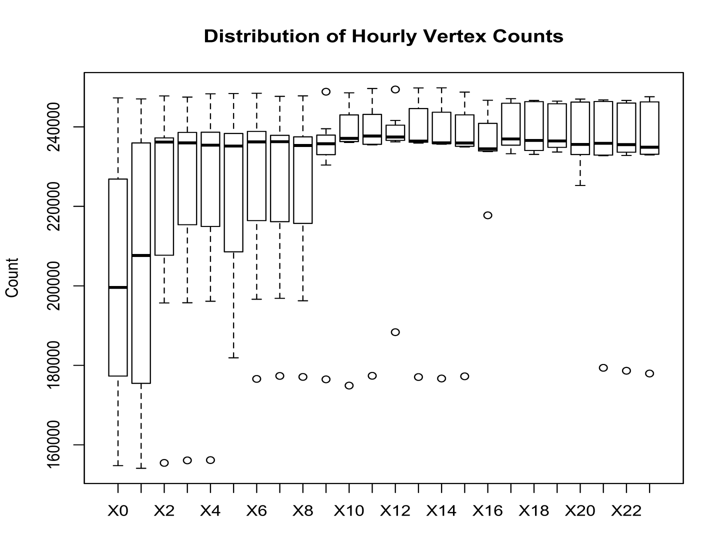
Host A IPv6 vs. Host A IPv4 (siblings)

Time-of-day Topology Effects

- NPS master's thesis: quantify extent that time of day impacts topology collection
- Stems from noticing the inability to compare between CAIDA cycles
- Less topology because of congestion?
- More topology because infrastructure is up?

Time-of-day Topology Effects

February, 2014 CAIDA Cycles



Basic Problem in Topology Research: Ground-Truth

Validation

- Yesterday we heard about the dangers of not validating topology measurements/tools and brittle inferences
- But obtaining ground-truth, validation is hard

Network Emulation

- Virtualization is cheap and easy today
- Even for routers (GNS3, Dynamips)
- Provides ability to easily spin up $O(100)$'s of routers with arbitrary connectivity running real router software
- NPS master's thesis: implementation creates different (random) topologies and policy in order to test tools
- If (tool, model) doesn't work in the lab, it won't work in the Internet

Deception

Yes, traceroute really is brittle

```
6 Episode.IV (206.214.251.1) 68.642 ms 67.307 ms 67.005 ms
7 A.NEW.HOPE (206.214.251.6) 65.986 ms 68.502 ms 68.708 ms
8 It.is.a.period.of.civil.war (206.214.251.9) 67.067 ms 70.139 ms 66.52
9 Rebel.spaceships (206.214.251.14) 70.214 ms 70.192 ms 71.622 ms
10 striking.from.a.hidden.base (206.214.251.17) 71.427 ms 74.206 ms
11 have.won.their.first.victory (206.214.251.22) 71.665 ms 70.434 ms 7
12 against.the.evil.Galactic.Empire (206.214.251.25) 69.218 ms 70.621
13 During.the.battle (206.214.251.30) 69.059 ms 68.931 ms 69.981 ms
14 Rebel.spies.managed (206.214.251.33) 77.247 ms 72.757 ms 77.61
15 to.steal.secret.plans (206.214.251.38) 71.224 ms 71.164 ms 69.543
16 to.the.Empires.ultimate.weapon (206.214.251.41) 68.744 ms 68.824

17 the.DEATH.STAR (206.214.251.46) 72.316 ms 74.551 ms 66.354 ms
```

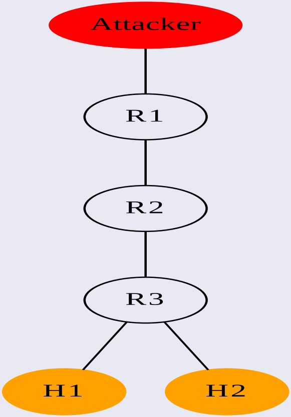
- How'd they do this?
- Can we be more formal?

Topology Deception

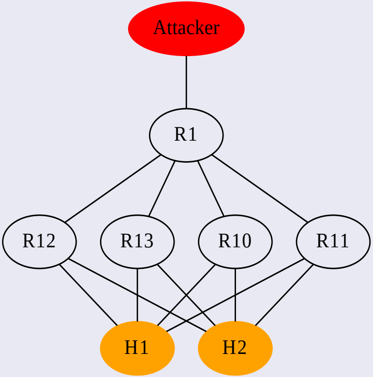
Sardine: Topology Deception (w/ Lt. Sam Trassare)

- Take inspiration from military deception (e.g., radar)
- Rather than block topology probes, return modified responses that cause adversary to infer a false topology:
  - Continuum: random vs. crafted responses
  - Graph theory: make weakest portion of topology appear to be most robust
- Keep adversary in collection rather than operational phase.
- Confuse adversary into believing least resilient portion of network is most robust.

Sardine Example



True Topology, Vulnerable Links



Faked Topology, False Links, Missing Links

Outstanding Topology Deception Work

Building Prototype:

- Linux-based router using `libnetfilter_queue`
- Redirect messages to userspace program for more complicated packet manipulation and reinjection
- Attempted implementation in SDN

Outstanding Questions:

- Maintaining consistency with multiple network ingresses
- What false topology to present?
- How to prevent detection of sardine?
- Performance issues

Labrea

Labrea Tarpit

- “Sticky” honeypot: listen for *unanswered* ARP requests on local segment, respond.
- Labrea implemented in Linux iptables
- W/ Lt. Lance Alt, have developed `degreaser`, an opensource tool to fingerprint and detect Labrea
- With Internet-wide scanning, we find instances of Labrea tarpitting in the wild
- Measurement researchers need to understand the impact of entities *actively* trying to deceive us.