IPv6 Topology Mapping

Robert Beverly[*], Ram Durairajan[†], Justin Rohrer[*], David Plonka[‡]

[*]Naval Postgraduate School
[†]University of Oregon
[‡]Akamai

April 20, 2018

Context

This talk:

- Really a bunch of slides for informal discussiom
- Planned IMC 2018 submission
- Looking for feedback/suggestions, especially on:
  - Motivation
  - Analysis/metrics/result interpretation

Problem

What is the topology of the IPv6 Internet?

Motivation:

- Understanding Internet topology is important (CDNs, security, resilience, diagnostics, research)
- IPv6 increasingly important (e.g., by metrics of traffic, enabled sites, edge deployment)
- But, do we have a good understanding of the IPv6 topology?

Prior Work

State-of-the-art:

- CAIDA has been collecting IPv6 topology maps for 10 years
- Essentially replicates their IPv4 probing methodology
- For every IPv6 prefix in the global BGP table*:
    - Use scamper (active measurement using ICMP6-Paris)
    - Traceroute to `::1` in prefix
    - Traceroute to random host in prefix
    - Takes approximately 9 hours for $< 100k$ targets (!)
- *March 5, 2018: probed 98,120 unique destinations; 1,782 destinations probed more than once (one probed six times)

Coverage/Completeness/Efficiency of state-of-art IPv6 topology maps has not been evaluated

Prior Work

State-of-the-art:

- CAIDA has been collecting IPv6 topology maps for 10 years
- Essentially replicates their IPv4 probing methodology
- For every IPv6 prefix in the global BGP table*:
  - Use scamper (active measurement using ICMP6-Paris)
  - Traceroute to `::1` in prefix
  - Traceroute to random host in prefix
  - Takes approximately 9 hours for $< 100k$ targets (!)
- *March 5, 2018: probed 98,120 unique destinations; 1,782 destinations probed more than once (one probed six times)

Coverage/Completeness/Efficiency of state-of-art IPv6 topology maps has not been evaluated

A different approach for IPv6?

Strategies for increasing coverage:

- Probe more destinations
- Probe faster
- Select better destinations

Probing faster:

RFC4443, §2.1.1: "*an IPv6 node MUST limit the rate of ICMPv6 error messages it originates*"

Assertion:

Existing tools/techniques ill-suited to IPv6 active topology mapping

A different approach for IPv6?

Strategies for increasing coverage:

- Probe more destinations
- Probe faster
- Select better destinations

Probing faster:

RFC4443, §2.1.1: "*an IPv6 node MUST limit the rate of ICMPv6 error messages it originates*"

Assertion:

Existing tools/techniques ill-suited to IPv6 active topology mapping

A different approach for IPv6?

Our approach:

1. Explore use of recent randomized, high-speed active topology probing techniques (Yarrp)
2. Develop IPv6 version of Yarrp
3. Evaluate efficacy of various hitlists, targets, and protocols
4. Goal: Produce the most complete IPv6 topology currently available

Yarrp

Yarrp: "Yelling at Random Routers Progressively" (IMC2016)

- `https://www.cmand.org/yarrp/`
- Uses a block cipher to **randomly permute** the $\langle IP, TTL \rangle$ domain
- Is **stateless**, recovering necessary information from replies
- By randomly spreading probes, permits **fast** Internet-scale active topology probing
  - (Runs $> 300kpps$, discovers $> 0.5M$ ints in $<$ 10min from single VP)

Hypothesis: Yarrp-mapping of the IPv6 Internet will suffer less rate-limiting, even at higher probing rates

CMAND

Yarrp

Yarrp: "Yelling at Random Routers Progressively" (IMC2016)

- `https://www.cmand.org/yarrp/`
- Uses a block cipher to **randomly permute** the $\langle IP, TTL \rangle$ domain
- Is **stateless**, recovering necessary information from replies
- By randomly spreading probes, permits **fast** Internet-scale active topology probing
  - (Runs $> 300kpps$, discovers $> 0.5M$ ints in < 10min from single VP)

Hypothesis: Yarrp-mapping of the IPv6 Internet will suffer less rate-limiting, even at higher probing rates

Yarrp Features Update

Lots of development since the 2016 IMC yarrp-0.1.

`yarrp-0.4:`

- TCP (SYN or ACK), UDP, or ICMP probes
- IPv4/IPv6
- Linux and BSD
- “Fill mode”
- Decoupled probing / receiving
- Biased probing
- Better usability / documentation

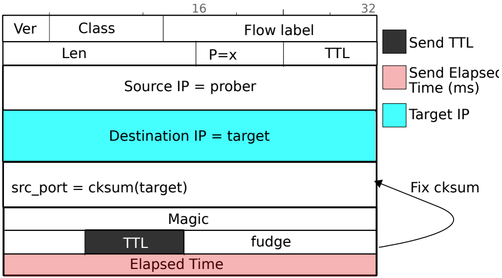
IPv6: Encoding State (new)

RFC4443, §3.3: ICMP6 TTL exceeded quotation includes "*As much of invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU*"

- In contrast to IPv4 which only guarantees 28B quote
- Significantly simplifies encoding – place in payload!

Yarrp6

IPv6: Encoding State



- Maintain constant transport (TCP/UDP/ICMP) header fields
- Encode TTL, elapsed time in transport payload
- Use 2B of "fudge" to correct payload so that checksum is correct

Fill Mode

Yarrp is stateless

- Must select TTL range (*maxTTL*) (potentially missing hops)
- Don't know when to stop probing (potentially wasting probes)
- (can be beneficial, as we discover hops beyond gap limit)

Fill mode

For response to a probe with TTL=h, probe with TTL=$h+1$ iff $h \geq maxTTL$. (not random, but uncommon and at path tail)

Win/win efficiency gain: Allows us to lower the *maxTTL* (less wasted probing), without missing hops.

SPECK

SPECK

- Use SPECK lightweight block cipher
- Faster, and supports intermediate block lengths (e.g., 48 and 96 bits – useful for IPv6-wide scanning)

Target Selection

- Must select IPv6 target addresses
- For 128-bit IPv6 address:
  - How to choose prefix? (Upper 64 bits)
  - How to host identifier? (Lower 64 bits)

Choosing prefixes

Basic strategies

- **Uniform:** probe all possible prefixes (generally infeasible)
- **Random:** chose prefixes at random (IPv6 space is too sparse)
- **BGP:** Select prefixes in global BGP table (does not capture subnetting)
- **Hitlists:** Select prefixes based on hitlists (utility/comparison of hitlists has not been previously explored in literature)
- **Generative:** Build a model of how prefixes are allocated using seeds of known addresses, generate new candidate prefixes (has not been previously explored in literature)

Hitlists

Name	Size	Method
CAIDA	93,894	BGP-derived
Fiebig	97,746	rDNS
Rapid7	196,012	fDNS
CDN Clients	372,930	Anonymous aggregates
DNS-DB	Varies	Farsight
6gen	Varies	Generative

- Lots of recent work on IPv6 hitlist/target generation
- No work to compare/understand these hitlists!
- Note, many hitlists have many addresses within same /64
  - We reduce these to unique /48s
  - (unlikely to produce interesting topology results)

Hitlists

Fiebig

- Intuition: leverage rDNS, i.e., IPv6 PTR records
- Recall, IPv6 PTR records are in the `ip6.arpa` namespace. Text hex nibbles separated by periods.
- e.g., `0.0.1.0.0.0.0.0.0.0.0.0.1.0.0.0.1.2.c.0.7.0.f.1.0.7.4.0.1.0.0.2.ip6.arpa. 3600 IN PTR ralph.rbeverly.net.`
- Trick: recent DNS standard specifies that resolvers respond to partial PTR queries with:
  - NXDOMAIN: no record and nothing under in the tree
  - NOERROR: children in the tree
- Permits (efficient) enumeration of the namespace
- Assumes providers populate the portion of the ip6.arpa hierarchy they are authoritative for

Hitlists

Rapid7

- scans.io performs regular internet-wide IPv4 service scans (e.g., web servers)
- Hitlist generated from AAAA queries for all names discovered

Hitlists

CDN clients

- Obtained in cooperation with Akamai
- Contains "anonymous aggregates" – prefixes based on the IPv6 addresses of clients that query the Akamai CDN platform
- Grouped in prefix aggregates such that the prefix is more specific than the BGP prefix, but has enough clients within it so that they remain anonymous

Hitlists

6gen

- Murdock et al. , IMC 2017: “*Target Generation for Internet-wide IPv6 Scanning*”
- Takes a set of input addresses (seed), attempts to learn the addressing structure
- Generates candidate addresses

Hitlists

Coverage distribution comparison

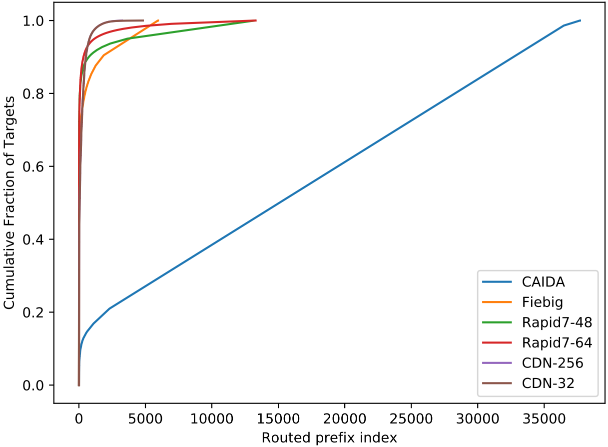


Figure: Cumulative targets per routed prefix

Hitlists

Coverage distribution comparison

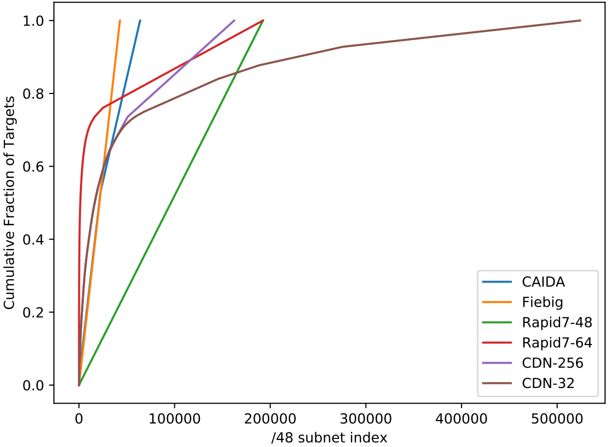


Figure: Cumulative targets per routed /48

Hitlists

Coverage distribution comparison

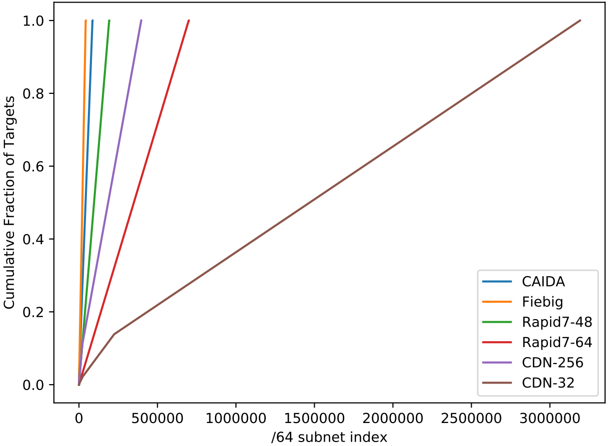


Figure: Cumulative targets per routed /64

Host identifier

Explored two strategies

- Probe `::1`
- Probe `1234:5678:1234:5678`

Initial Active IPv6 Probing Results

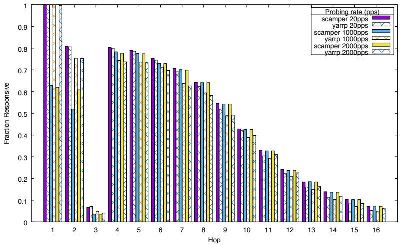
Vantage Points:

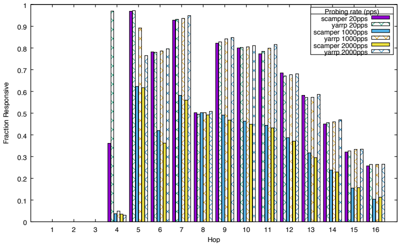
- NPS
- UOregon
- Swiss IXP

Probed:

- Different speeds
- Different hitlists
- Different prefixes/hosts
- Different protocols

Benefit of randomization





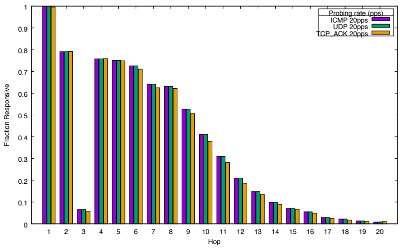
NPS UOregon

- Same targets, same vantage point
- Yarrp outperforms scamper, especially near source
- Clearly some hops exhibit different rate-limiting behavior (`lax-agg6-lax-hpr3-100g.cenic.net.` and (`3.be-1.uonet9-gw.uoregon.edu.`))

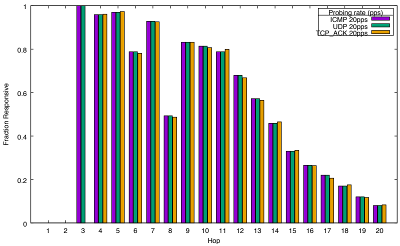
Different Router Behaviors

- Clearly some hops exhibit different rate-limiting behavior
- `3.be-1.uonet9-gw.uoregon.edu.`
- Queried owner, is a Cisco ASR9000
- "*It's going to be replaced with a Juniper MX10003 some time in the next month or so. We're not doing anything special beyond ACLs and Netflow sampling on those router interfaces*"

Probe Type



NPS



UOregon

Take-away:

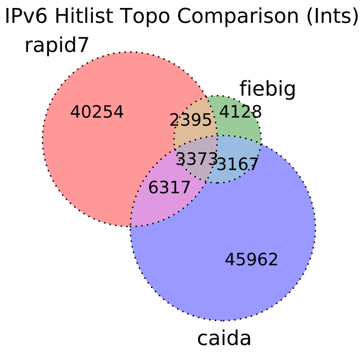
- Marginal difference; ICMP/UDP most innocuous

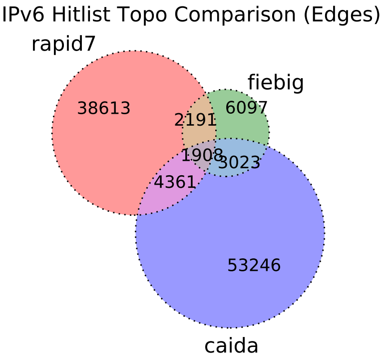
Host identifier

ICMP6 response types

	CDN		
type/code	**::1**	**1234:5678**	**Fiebig**
Hop lim	3598838 (98.1%)	3530329 (98.1%)	380089 (95.8%)
No route	26728 (0.7%)	26039 (0.7%)	2454 (0.6%)
Adm prohib	23595 (0.6%)	21080 (0.6%)	1696 (0.4%)
Addr unrch	12158 (0.3%)	14781 (0.4%)	2631 (0.7%)
Port unrch	5250 (0.1%)	1045 (0.0%)	9084 (2.3%)
Reject rte	2577 (0.1%)	6279 (0.2%)	818 (0.2%)

Hitlist evaluation

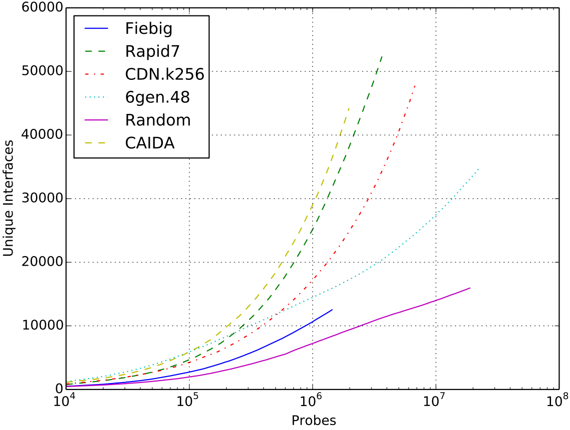




Take-away:

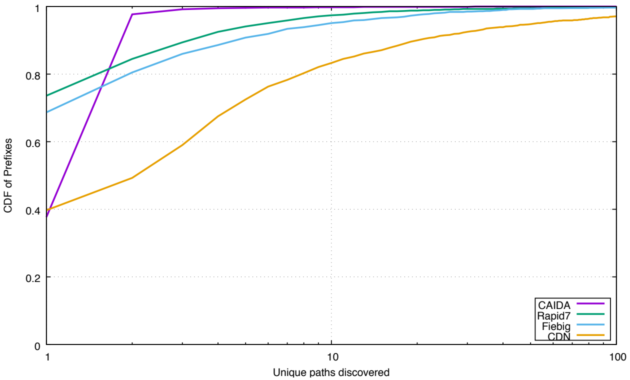
- Missing topology in state-of-the-art
- Hitlists are complementary

Hitlist Power



- Again, very complementary behavior

Subnetting



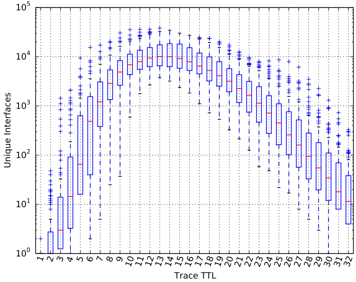
- How many distinct paths to different targets within same routed prefix?
- Use as a basic proxy for amount of subnetting

Outstanding questions/work:

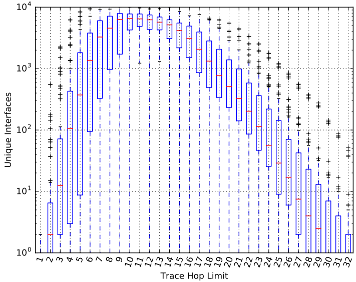
- How to best evaluate hitlists (counts, graph metrics, subnetting, etc)?
- Characterize discovered topology (ASes, edge/core, etc)
- Resolve aliases, determine how many new routers discovered
- Combining hitlists
- Creating the definitive “topology-of-record” for IPv6

Path diameters

Distribution of interfaces over all VPs



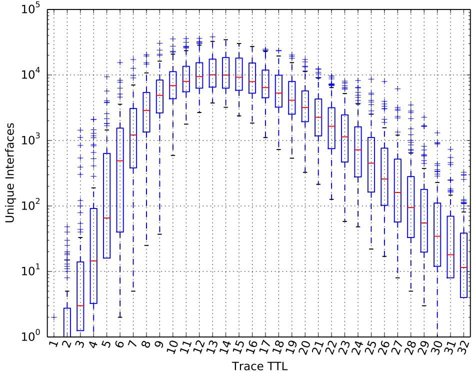
IPv4



IPv6

- Interesting skew toward smaller diameters in IPv6
- Likely due to tunneling (e.g., Hurricane Electric)
- And relative size of big IPv6 ASes

Biased Probability



Given the distribution of routers vs. TTL

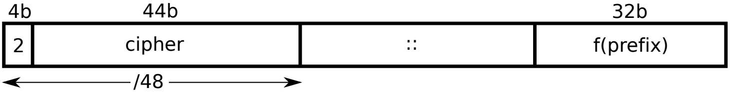
- Use different probability distributions to bias the search

Current Option Summary

```
OPTIONS:
  -i, --input               Input target file
  -o, --output              Output file (default: output.yrp)
  -c, --count               Probes to issue (default: unlimited)
  -t, --type                Type: ICMP, TCP_SYN, TCP_ACK, UDP, ICMP6, UDP6, TCP6_SYN, TCP6_ACK (defa
  -r, --rate                Scan rate in pps (default: 10)
  -m, --maxttl              Maximum TTL (for ip input list only)
  -v, --verbose             verbose (default: off)
  -F, --fillmode            Fill mode maxttl (default: 0)
  -s, --sequential          Scan sequentially (default: random)
  -n, --neighborhood        Neighborhood TTL (default: 0)
  -b, --bgp                 BGP table (default: none)
  -S, --seed                Seed (default: random)
  -p, --port                Transport dst port (default: 80)
  -T, --test                Don't send probes (default: off)
  -Q, --entire              Entire IPv4/IPv6 Internet (default: off)
  -I, --interface           Network interface (required for IPv6)
  -G, --dstmac              MAC of gateway router (default: auto)
  -M, --srcmac              MAC of probing host (default: auto)
```

Internet-Wide Probing

Forming an IPv6 target



- Use 48-bit Speck block cipher
- First 4-bit nibble fixed (IANA allocation of `0x2`)
- Add 44 bits of cipher to form /48 prefix to probe
- Lower 32 bits a deterministic function of /48 prefix
- Remaining 4 bits of cipher determine TTL (1-16)