Randomized High-Speed Active Topology Discovery

Robert Beverly

Naval Postgraduate School

September 22, 2016

Akamai Seminar

Outline

1. Introductions
2. Background
3. Methodology
4. Results
5. Conclusions

CMAND Lab @ NPS

Naval Postgraduate School (NPS)

- Navy's Research University, Monterey, CA
- $\simeq$1500 students, military officers, DoD civilians

Center for Measurement and Analysis of Network Data

- 3 NPS professors, 2 NPS staff
- 1 PhD student, rotating cast of $\sim$5-8 Master's students
- Collaborators: CAIDA, ICSI, MIT, Akamai, Cisco, Verisign, USNA
- Funding: NSF, DHS

Focus:

- Large-scale network measurement and data mining
- Network architecture and security

CMAND Lab @ NPS

Naval Postgraduate School (NPS)

- Navy's Research University, Monterey, CA
- $\simeq$1500 students, military officers, DoD civilians

Center for Measurement and Analysis of Network Data

- 3 NPS professors, 2 NPS staff
- 1 PhD student, rotating cast of $\sim$5-8 Master's students
- Collaborators: CAIDA, ICSI, MIT, Akamai, Cisco, Verisign, USNA
- Funding: NSF, DHS

Focus:

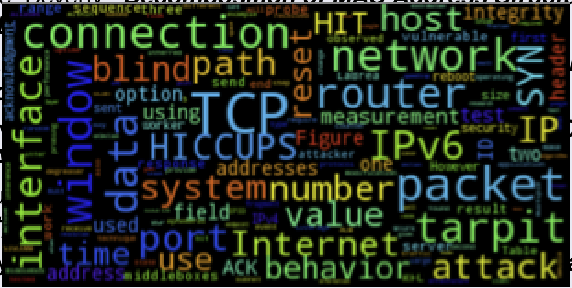
- Large-scale network measurement and data mining
- Network architecture and security

Some Recent Publications:

1. Beverly, "***Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery***," in ACM IMC 2016 (to appear)
2. Martin, Rye, Beverly, "***Decomposition of MAC Address Structure for Granular Device Inference***," in ACSAC 2016 (to appear)
3. Rohrer, LaFever, Beverly, "***Empirical Study of Router IPv6 Interface Address Distributions***," in IEEE IC 2016
4. Luckie, Beverly, Wu, Allman, Claffy, "***Resilience of Deployed TCP to Blind Off-Path Attacks***," in ACM IMC 2015 (best paper)
5. Beverly, Luckie, Mosley, Claffy, "***Measuring and Characterizing IPv6 Router Availability***," in PAM 2015
6. Alt, Beverly, Dainotti, "***Uncovering Network Tarpits with Degreaser***," in ACSAC 2014
7. Craven, Beverly, Allman, "***A Middlebox-Cooperative TCP for a non End-to-End Internet***," in ACM SIGCOMM 2014

Some Recent Publications:

1. Beverly, "***Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery***," in ACM IMC 2016 (to appear)
2. Martin, Rye, Beverly, "***Decomposition of MAC Address Structure for Granular***
3. Rohrer, La ***rface Address Distributio***
4. Luckie, Be ***CP to Blind Off-Path A***
5. Beverly, Lu ***IPv6 Router Availabilit***
6. Alt, Beverly ***aser***," in ACSAC 2014



7. Craven, Beverly, Allman, "***A Middlebox-Cooperative TCP for a non End-to-End Internet***," in ACM SIGCOMM 2014

Long history of collaboration with Akamai:

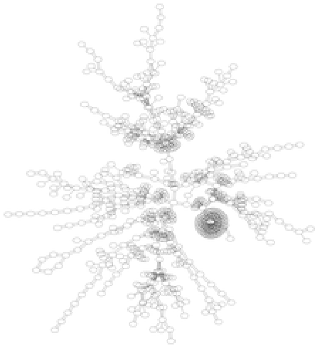
1. Beverly, Berger, "***Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure via Active Fingerprinting***," in PAM 2015
2. Berger, Weaver, Beverly, Campbell, "***Internet Nameserver IPv4 and IPv6 Address Relationships***," in IMC 2013
3. Bauer, Beverly, Berger, "***Measuring the State of ECN Readiness in Servers, Clients, and Routers***," in IMC 2011
4. Beverly, Berger, Xie, "***Primitives for Active Internet Topology Mapping***," in IMC 2010
5. Beverly, Berger, Hyun, claffy, "***Understanding the Efficacy of Deployed Internet Source Address Validation Filtering***," in IMC 2009

Outline

1. Introductions
2. Background
3. Methodology
4. Results
5. Conclusions

Internet Topology

Long-standing question: *What is the topology of the Internet?*



Why care (I)?

- **Researchers:** network modeling, network science, routing protocol validation, new architectures, Internet evolution, etc.
- **CDNs:** optimize content delivery over a time-varying graph with dynamic workloads
- **Network management:** understand traffic paths, diagnose faults and performance problems
- **Policy makers:** understand provider interconnection, broadband availability, consumer choice, congestion points, differentiated service
- **Security:** critical infrastructure protection, detecting routing hijacks

Why care (II)?

"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." – DHS

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.

Topology Mapping Challenges

Difficult to answer – Internet is:

- A large, complex distributed system (organism)
- Non-stationary (in time)
- Difficult to observe, multi-party (information hiding for scalability and competitive reasons)
- Poorly instrumented (not part of original design)

⇒ Today, Internet topology remains poorly understood (at interface, router, AS, or organization level)

Topology Mapping Challenges

Difficult to answer – Internet is:

- A large, complex distributed system (organism)
- Non-stationary (in time)
- Difficult to observe, multi-party (information hiding for scalability and competitive reasons)
- Poorly instrumented (not part of original design)

$\Rightarrow$ Today, Internet topology remains poorly understood (at interface, router, AS, or organization level)

How can we map the Internet topology?

Mapping Approaches:

- Passive inference vs. active probing
- Fixed vs. opportunistic vantage points
- Directed vs. uniform probing

Continuous Topology Measurement

- Archipelago (CAIDA), iPlane (UW)
- Ark IPv4 probing strategy:
  - IPv4 space divided into /24's; partitioned across monitors
  - From each /24, select a single address at random to probe
  - Probe == Scamper [L10]; record router interfaces on forward path
  - A "cycle" == probes to all routed /24's

Active Topology Probing

- **Years** (and years) of prior work on Internet-scale topology probing
- Current production systems take several **days** from 100's of vantage points to gather a coarse-granularity network map
- Topology "snapshots" are a misnomer! – network can **change** during probing
- Difficult to predict path changes and probe proportionally (DTrack)

It's 2016:

- Why can't we traceroute to *every* IPv4 destination *quickly*?
- e.g., $O(minutes)$?
- (The ZMap[a] and Masscan[b] folks can do it – why can't we?)

[a] Z. Durumeric et al., 2013
[b] R. Graham, 2013

Active Topology Probing

- **Years** (and years) of prior work on Internet-scale topology probing
- Current production systems take several **days** from 100's of vantage points to gather a coarse-granularity network map
- Topology "snapshots" are a misnomer! – network can **change** during probing
- Difficult to predict path changes and probe proportionally (DTrack)

It's 2016:

- Why can't we traceroute to *every* IPv4 destination *quickly*?
- e.g., $O(minutes)$?
- (The ZMap[a] and Masscan[b] folks can do it – why can't we?)

[a] Z. Durumeric et al., 2013
[b] R. Graham, 2013

Existing traceroute-style approaches:

- Maintain **state** over outstanding probes (identifier, origination time)
- Are **sequential**, probing all hops along the path. Any parallelism limited to a window of outstanding paths being probed.

Implications:

- **Concentrates load:** along paths, links, routers (potentially triggering rate-limiting or IDS alarms)
- Production systems probe **slowly**

Existing traceroute-style approaches:

- Maintain **state** over outstanding probes (identifier, origination time)
- Are **sequential**, probing all hops along the path. Any parallelism limited to a window of outstanding paths being probed.

Implications:

- **Concentrates load:** along paths, links, routers (potentially triggering rate-limiting or IDS alarms)
- Production systems probe **slowly**

Outline

1. Introductions
2. Background
3. Methodology
4. Results
5. Conclusions

Yarrp

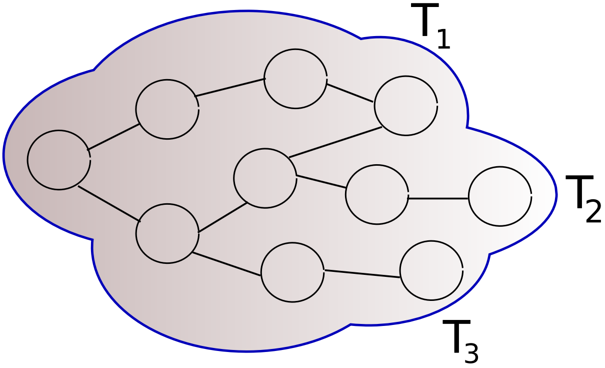
Yarrp: “Yelling at Random Routers Progressively”
(To appear, ACM Internet Measurement Conference, Nov, 2016)

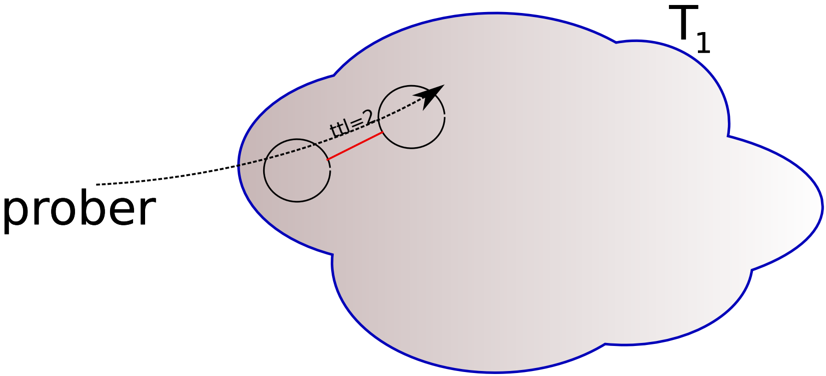
Takes inspiration from ZMap:

- Uses a block cipher to **randomly permute** the $< IP, TTL >$ space
- Is **stateless**, recovering necessary information from replies
- Permits **fast** Internet-scale active topology probing (even from a single vantage point)

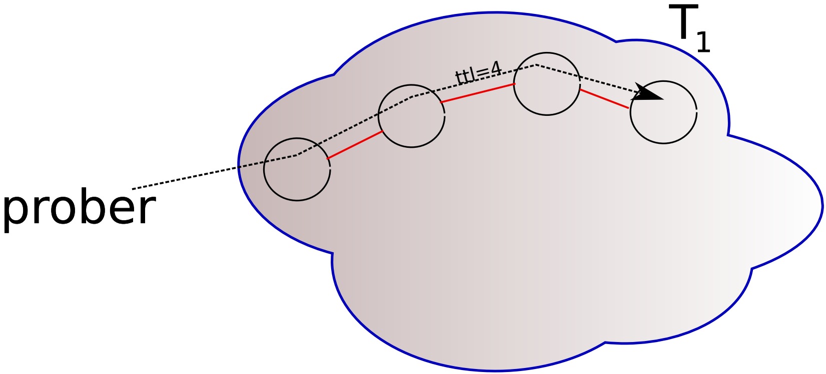
Example Topology

prober

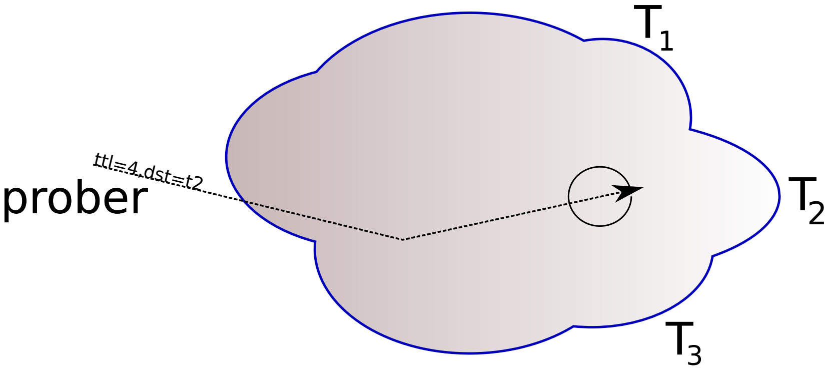




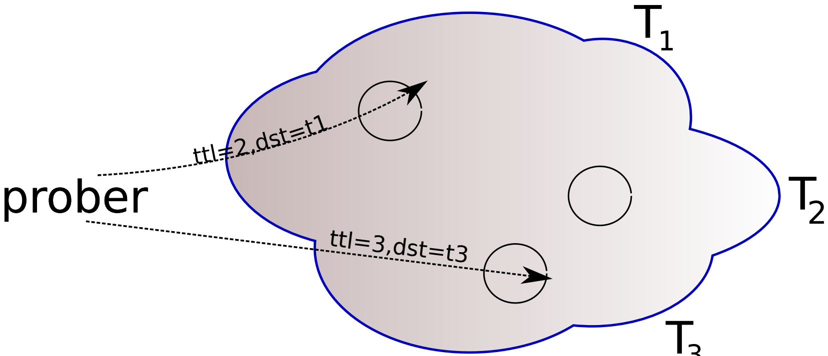
Traditional traceroute sends probes with incrementing TTL to destination T_1



... continuing until finished with T_1 (reach destination or gap limit).
Prober must maintain state,
while traffic is concentrated on *prober* $\rightsquigarrow$ T_1 path

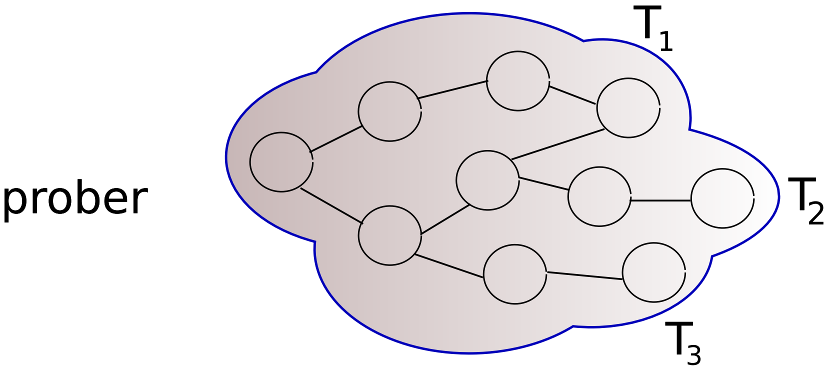


In contrast, Yarrp iterates through randomly permuted $< \textit{Target}, \textit{TTL} >$ pairs



In contrast, Yarrp iterates through randomly permuted $< \textit{Target}, \textit{TTL} >$ pairs

Inferred Topology



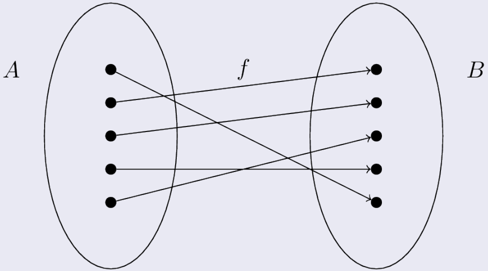
Finally, stitch together topology. Requires state and computation, but decoupled (off-line after probing completes).

Challenges:

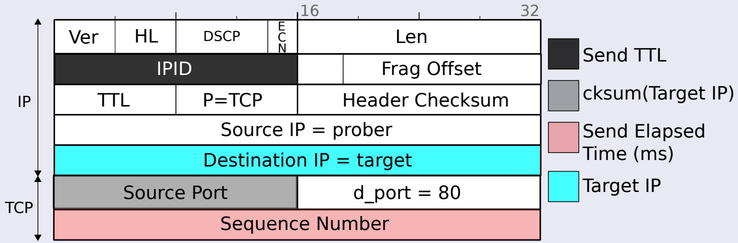
1. Creating the random probing order
2. Map responses to the originating probe's destination, TTL, and origin time
3. Knowing when to stop probing a path (max TTL)
4. Handling typical load-balancing issues

Pseudo-random Probing Order

- We use RC5 block cipher with 32-bit block size
- Encrypt $i = 0, \ldots, 2^{32} - 1$ with key k to obtain /24's and TTLs:
  - $C_i = RC5_k(i)$
  - /24 = $C_i[0:23]$
  - TTL = $C_i[24:31]$
  - Least-significant octet: $f(C_i[0:23])$



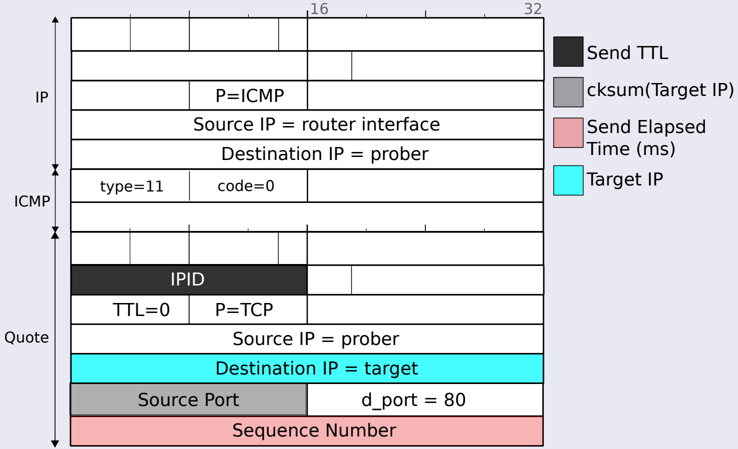
Encoding State



- IPID = Probe's TTL
- TCP Source Port = cksum(Target IP destination)[a]
- TCP Seq No = Probe send time (elapsed ms)
- Per-flow load balancing fields remain constant (ala Paris)
- Assume routers echo only 28B of expired packet

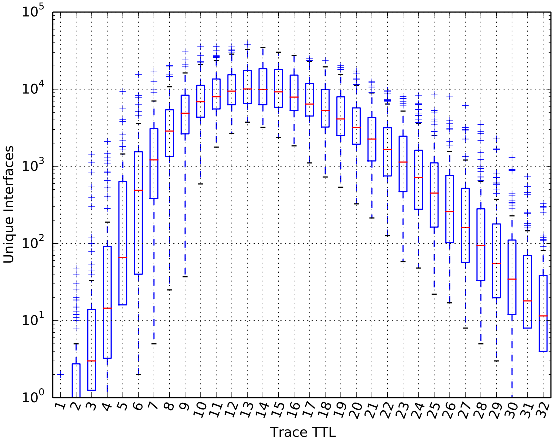
[a]Malone PAM 2007: ≈2% of quotations contained modified destination IP

Recovering State



ICMP TTL exceeded replies permit recovery of: target probed, originating TTL (hop), and responding router interface at that hop.

Distribution of unique interfaces discovered vs. TTL for all Ark monitors, one Ark topology probing cycle



- Problem: knowing when to stop
- Little discoverable topology past TTL=32
- $\Rightarrow$ limit $< IP, TTL >$ search space to $TTL \leq 32$

Implementation

Yarrp Implementation

- C++ ~2,500 SLOC
- Independent send and receive threads
  - Send thread uses raw sockets
  - Receive thread uses libpcap
- Portable to variety of UNIX-like platforms
- Publicly available:
  `https://www.cmand.org/yarrp`

Decoupling Probing from Reconstruction

Yarrp Data

- Receive thread runs independently
- Recovers state and writes responses
- Because probing is randomized, replies are unordered:

```
# yarrp $Id: yarrp.cpp 40 2016-01-02 18:54:39Z rbeverly $
# Started: Tue May 10 12:52:41 2016
# Source: 18.26.2.84, Count: 0 Rate: 4000
# Rand: 1 Nbrh: 0 Entire: 0 BGP: bgptable.20160510.txt.gz TraceType: 3
# Input IPlist: /home/rbeverly/c004710.san-us.targets MaxTTL: 16
# target, sec, usec, type, code, ttl, hop, rtt, ipid, psize, rsize, rttl, rtos
109.112.178.108, 1462899605, 97182, 11, 0, 8, 198.71.47.61, 22, 0, 40, 56, 248, 0
75.227.91.50, 1462899605, 97299, 11, 0, 9, 4.68.110.82, 5, 0, 40, 56, 246, 0
150.243.54.100, 1462899605, 97418, 11, 0, 6, 18.192.7.2, 1, 2310, 40, 96, 250, 0
179.130.181.73, 1462899605, 98230, 11, 0, 14, 200.220.224.253, 206, 10160, 40, 56, 235, 72
42.97.123.149, 1462899605, 99366, 11, 0, 11, 64.57.20.146, 54, 0, 40, 56, 245, 0
198.48.67.42, 1462899605, 100550, 11, 0, 1, 18.26.0.2, 10, 55674, 40, 56, 255, 0
104.3.115.120, 1462899605, 100666, 11, 0, 10, 12.122.130.170, 50, 25157, 40, 168, 240, 0
84.106.41.175, 1462899605, 100953, 11, 0, 13, 84.116.195.246, 133, 48736, 40, 56, 241, 0
76.216.172.133, 1462899605, 101268, 11, 0, 15, 12.122.30.30, 83, 23223, 40, 172, 239, 0
74.150.100.227, 1462899605, 102383, 11, 0, 10, 68.85.184.198, 8, 10, 40, 56, 246, 192
108.76.185.84, 1462899605, 102395, 11, 0, 14, 12.122.30.25, 78, 28971, 40, 172, 242, 0
155.198.102.65, 1462899605, 103470, 11, 0, 11, 62.40.98.76, 83, 0, 40, 56, 245, 0
```

Decoupling Probing from Reconstruction

Topology Reconstruction

- `yrp2warts.py`: assembles unordered Yarrp responses into series of binary warts-formatted traces
- Currently unoptimized, single-threaded
- 6.8M destinations `.yrp` → `.warts` in 668 sec

```
traceroute from 18.26.2.84 to 190.144.172.20
 1  18.26.0.2  1.000 ms
 2  128.30.0.245  1.000 ms
 3  128.30.13.5  1.000 ms
 4  18.4.7.1  4.000 ms
 5  18.192.2.1  1.000 ms
 6  18.192.7.2  1.000 ms
 7  207.210.143.109  1.000 ms
 8  192.5.89.21  1.000 ms
 9  192.5.89.222  6.000 ms
10  198.71.46.174  24.000 ms
11  200.0.207.9  36.000 ms
12  200.0.204.6  84.000 ms
13  200.0.204.182  147.000 ms
```

Optimizations

- Base Yarrp requires no state
- (Must reconstruct traces, but that's an offline local process)
- If we're willing to maintain some space, we can optimize: Time Memory Trade Off
  1. Probe only routed destinations (radix trie BGP RIB)
  2. Avoiding repeated re-discovery of prober's local neighborhood (state over small number of interfaces near prober)
- (See paper for full details)

Outline

1. Introductions
2. Background
3. Methodology
4. Results
5. Conclusions

Ethical Concerns

- High-speed probing increases chance traffic perceived as abusive
- Yarrp sends TCP ACK probes (less abusive than ZMap's SYNs)
- Random probing order avoids overloading networks
- Stateless nature implies multiple probes with different TTLs may reach a single destination
- We follow good "Internet citizenship" guidelines:
  - Coordinated with local network admins
  - Informative web page at address of prober
  - DNS PTR record indicates research nature
  - Provide links to opt-out
- In our $\leq$ 60 min Yarrp runs, we received no abuse reports or opt-outs

Yarrp Speed

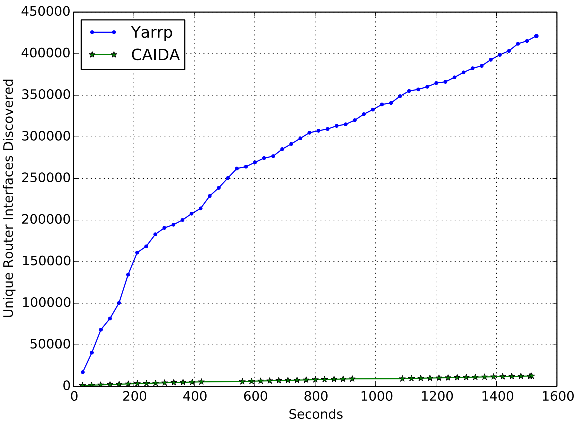
Calibration:

- Examine an Ark topology cycle (probe one address in all routed /24's) from April, 2016.
- Sent $\approx$ 11M traceroutes from 37 monitors over 31 hours
- Discovered $\approx$1M router interfaces, $\approx$2M links

Yarrp:

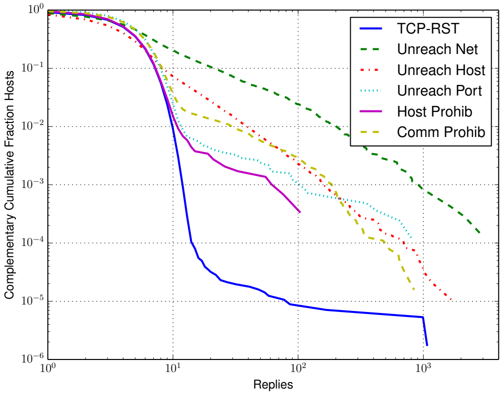
- Sent 10M probes in $\approx$100 sec ($\approx 100Kpps$))
  - Found 178,453 unique router interfaces
- Discover >400,000 interfaces in <30 min from a single vantage

Yarrp vs. Ark



- Well-provisioned university vantage point
- Yarrp running on KVM (1 core @ 2.27GHz) at 100*kpps*, 52% CPU
- Yarrp: $\approx$ 280 unique router interfaces / sec
- Ark: $\approx$ 8 unique router interfaces / sec

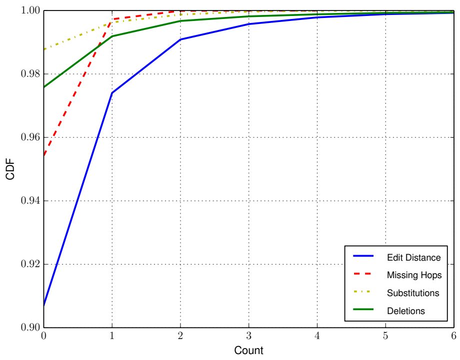
Non-TTL Exceeded Replies



- Yarrp's TCP probing elicits a variety of responses
- ~95K ICMP Host Unreach, ~63K ICMP Communication Prohib
- Received ~1.2M TCP RST packets
- But, 99.1% of hosts sending a RST sent ≤ 10
- (3 IPs in Wanadoo send majority)

Short-Lived Dynamics

Application: Rapid Snapshots



- 67k targets, three Yarrp snapshots in succession (same k)
- Examine edit distance between S_1 and S_2
- 91% of paths identical, 6% have single hop difference
- 4% of have 1 hop differences due to missing hops, 1% substitutions

Short-Lived Dynamics

Example, probe toward ASN 262316

```
... 18.192.9.2 4.53.48.97 4.69.144.80 4.69.144.80 4.26.0.166 201.48.50.161 201.48.50.154   201.48.
... 18.192.9.2 207.210.142.229 198.71.47.57 * 67.16.148.6    201.48.50.161 187.115.214.189 187.115
... 18.192.9.2 38.104.186.185 154.54.30.41 154.54.47.30 154.54.11.110 64.210.21.110 213.155.131.23
```

Resulting AS path

```
3 3356 16735 28303
3 10578 11164 3549 16735 18881 4.172
3 174 3549 1299 25933 16735
```

- Confirmed BGP churn visible at routeviews
- Dynamics invisible to existing active topology probing systems

Outline

1 Introductions

2 Background

3 Methodology

4 Results

5 Conclusions

Future

Yarrp Enhancements

- **IPv6 Probing:**
  - Given vastly larger address space, Yarrp may enable gathering of more complete maps
  - Different IPv6 headers imply different encoding
  - But, full packet quotation in ICMP6 enables more flexibility
- **UDP Probing:**
  - TCP probing is known to be blocked more often and trigger more alerts
  - Encode timestamp into the length and checksum; create a payload to make checksum correct
- **ICMP Probing:**
  - Encode timestamp into identifier and sequence number; create payload s.t. each packet has same checksum

Future

Distributed Probing

- Use cryptographic permutation to divide probing among multiple vantage points
- Minimal communication overhead, distribute *key*, size of domain $|D|$, number of vantage points n, and vantage point id v. Then:

for $i \in |D|$ **do**
  $(ip, ttl) = E_{key}(i)$
  if $ip\%(n-1) == v$ **then**
    probe(ip, ttl)

- Speed scales linearly with n
- Given 100kpps and $n = 128$, traceroute to every routed IPv4 address in under 1 hour

Yarrp'ing the Internet

- New technique for rapid active topology discovery
- Redefine notion of a topology "snapshot"
- Demonstrate ability to detect short-lived dynamics
- Publicly available implementation

Thanks! – Questions?

`https://www.cmand.org/yarrp`