Detecting IXPs in traceroute paths

www.inspire.edu.gr/traIXroute

George Nomikos
FORTH – ICS, INSPIRE
gnomikos @ ics.forth.gr

Prof. Xenofontas Dimitropoulos
UoC / FORTH – ICS, INSPIRE
“... if and where an IXP was crossed.”

Transparency

Evolution

End-to-End paths Troubleshooting
Observing an IP address from an IXP prefix is not sufficient to infer an IXP crossing

1. Third-party IPs:

2. The available IXP prefix data may be:
   a) inaccurate, or
   b) could be used in other subnets
traIXroute

- Python 3
- Open Source
- GPLv3
- First build was released in May 2016 – v1.0
  - After PAM Conference in April, 2016 [1].
- Last version was released in February 2017 – v2.1.1

tralIXroute: Key Features

✓ A general purpose tool to detect IXP hops on-the-fly
✓ Exploits easily accessible IXP data
✓ Overcomes existing shortcomings
✓ Inter-operability with RIPE Atlas
✓ Remote Peering identification [1]
✓ Modular design and customization

Modular Design & Workflow

1. Input Data
   - Initialization

2. Sending Probes
   - IP Path

3. 3-Step Analysis
   - Methodology

4. Path Extraction
   - Output

Workflow

Modules
1. **IXP Memberships**
   - e.g. 198.32.118.24 – AS10310 – Equinix New York

2. **IXP Subnets**
   - e.g. 198.32.118.0/24 – Equinix New York

3. **RouteViews Prefix-to-AS mappings**
   - e.g. 64.233.160.0/24 – AS15169

---

**Initialization - Input Data**

Provided by:

- PeeringDB (PDB)
- Packet Clearing House (PCH)
- CAIDA based on RouteViews data
Data Accuracy & Validation

- **PDB** data are primarily self-reported by IXP and ISP operators.
- **PCH** is based on BGP Route Collectors (RCs) located in IXPs.

Based on the BGP dumps from **87 RCs on IXPs** operated by PCH
we validated the:

- **93.4%** of the IXP Membership data from **PDB**
- **92.1%** from **PCH**
• We send the probe to a certain destination using:
  o Traceroute
  
or
  o Scamper [1]

Methodology Overview

The IXP identification mechanism proceeds as follows:

• Step 1: Detect IXP IPs in traceroute paths based on IXP Membership data and/or prefixes

• Step 2: **Check the IXP membership** of the ASes adjacent to the observed IXP address(es)

• Step 3: Identify the **IXP crossing link**
Methodology Overview

• Step 1: Detect IXP IPs in traceroute paths based on IXP Membership and/or prefixes data

• Step 2: Check the IXP membership of the ASes adjacent to the observed IXP address(es)

• Step 3: Identify the IXP crossing link
We apply a sliding window of size 2 or 3 IP addresses.
Methodology – Step 1

- Does the IP address in the **middle** match an exact BGP router IP address from an IXP subnet?

**IXP Memberships and / or Prefixes**

e.g. Entry: 198.32.118.24 – AS10310 – Eq. NY
e.g. Prefix: 198.32.118.0/24 – Eq. NY
• Step 1: Detect IXP IPs in traceroute paths based on IXP Membership and/or prefixes data

• Step 2: **Check the IXP membership** of the ASes adjacent to the observed IXP address(es)

• Step 3: Identify the **IXP crossing link**
Methodology – Step 2

• Are the adjacent ASes members of the IXP?

Prefix-to-AS mappings
e.g. 64.233.160.0/24 – AS15169

IXP Memberships
e.g. 198.32.118.24 – AS10310 – Eq. NY
Methodology Overview

• Step 1: Detect IXP IPs in traceroute paths based on IXP Membership and/or prefixes data

• Step 2: Check the IXP membership of the ASes adjacent to the observed IXP address(es)

• Step 3: Identify the IXP crossing link
Methodology – Step 3

• Is the IXP link crossed before or after the IXP IP address?
  
  o Check when sufficient information about the ASes is available.
We propose **strong** and **weak** evidence rules.

**Strong**:

\[ IP^{bgp} \rightarrow AS \quad \text{IXP} \rightarrow IP^{inf} \rightarrow AS \quad IP^{bgp} \rightarrow AS \]

\[ \text{AS} \neq \text{AS} = \text{AS} \rightarrow a \]

**Weak**:

\[ IP^{bgp} \rightarrow AS \quad \text{IXP} \rightarrow IP^{prf} \rightarrow \text{IXP} \quad IP^{bgp} \rightarrow AS \]

\[ \text{AS} \rightarrow \text{IXP} \quad \text{AS} \rightarrow a \]
python3 traIXroute.py -asn -dns -rule -db probe -dest 109.110.48.142 -s

Imported 13 IXP Detection Rules from Rules.txt.
Loading from Database.
Imported 16 Reserved Subnets.
Extracted 0 IXP IPs and 0 IXP Subnets from additional_info.txt.
Extracted 17166 IXP IPs from PDB.
Extracted 3905 IXP IPs from PCH.
Extracted 469 IXP Subnets from PDB.
Extracted 371 IXP Subnets from PCH.
Extracted 17111 no dirty IXP IPs after merging PDB, PCH and additional_info.txt.
Extracted 1130 dirty IXP IPs after merging PDB, PCH and additional_info.txt.
Extracted 573 IXP Subnets after merging PDB, PCH and additional_info.txt.
traIXroute using scamper with default options.
traIXroute from union-tel.193.9.ru (94.228.193.9) to atlas-noc.podryad.tv (109.110.48.142).

<table>
<thead>
<tr>
<th></th>
<th>AS</th>
<th>IP Address</th>
<th>Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AS*</td>
<td>10.184.252.246 (10.184.252.246)</td>
<td>0.749 ms</td>
</tr>
<tr>
<td>2</td>
<td>AS48293</td>
<td>union-tel.192.166.ru (94.228.192.166)</td>
<td>0.957 ms</td>
</tr>
<tr>
<td>3</td>
<td>AS48293</td>
<td>union-tel.207.126.ru (94.228.207.126)</td>
<td>1.006 ms</td>
</tr>
<tr>
<td>4</td>
<td>(DE-CIX)-&gt;AS20485</td>
<td>frt01.transtelecom.net (80.81.194.117)</td>
<td>41.73 ms</td>
</tr>
<tr>
<td>5</td>
<td>AS20485</td>
<td>vvk06.transtelecom.net (217.150.59.102)</td>
<td>154.193 ms</td>
</tr>
<tr>
<td>6</td>
<td>AS20485</td>
<td>IP-Kozitskiy-gw.transtelecom.net (217.150.59.101)</td>
<td>149.7 ms</td>
</tr>
<tr>
<td>7</td>
<td>AS196949</td>
<td>Po1-20g.c65.vss.core.vl.podryad.tv (109.110.48.146)</td>
<td>150.148 ms</td>
</tr>
<tr>
<td>8</td>
<td>AS196949</td>
<td>109.110.48.141 (109.110.48.141)</td>
<td>153.489 ms</td>
</tr>
<tr>
<td>9</td>
<td>AS196949</td>
<td>atlas-probel.ripe.noc.podryad.tv (109.110.48.142)</td>
<td>155.008 ms</td>
</tr>
</tbody>
</table>

IXP Hops:
Rule: 1 -- 3) 94.228.207.126(AS48293) <--DE-CIX (DE,Frankfurt)---> 4) 80.81.194.117(AS20485)
Remote Peering:
Rule: 1 -- 3) 94.228.207.126 (AS48293,Europe,moscow,36.89ms) <---> DE-CIX (DE,Frankfurt)
Use Case: IXPs in traceroute paths

• Methodology
  o **31.8 million** probed paths collected from the CAIDA’s Ark measurement infrastructure*
  o **16 IXP detection rules**

*Data collected on January, 20th 2015

• Results
  o How often paths cross IXPs? **...17.4% – 23.6%**
  o How many IXPs are encountered per path? **...1 – 1.05**
  o Where is the IXP hop located? **...5.4 – 6.68 hop**
Conclusions

• **tralXroute**, an open source tool to identify IXP hops in IP paths
  o Fast & on-the-fly IXP detection
  o Inter-operability with RIPE Atlas & Remote Peering Jedi
  o More transparency and easier network troubleshooting
  o [www.inspire.edu.gr/tralXroute](http://www.inspire.edu.gr/tralXroute)

• Ongoing & Future work:
  o IPv6 support
  o Ground truth for validation
  o Additional IXP databases
  o More accuracy & functionality

Thank You!!!

gnomikos @ ics.forth.gr