traIXroute

Detecting IXPs in traceroute paths

www.inspire.edu.gr/traIXroute

George Nomikos

FORTH – ICS, INSPIRE

gnomikos @ ics.forth.gr

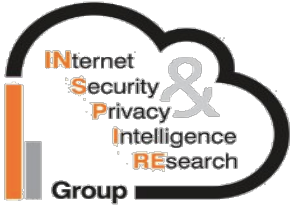
Prof. Xenofontas Dimitropoulos

UoC / FORTH – ICS, INSPIRE

FORTH

FOUNDATION FOR RESEARCH AND TECHNOLOGY - HELLAS

INSTITUTE OF COMPUTER SCIENCE



“… if and where an IXP was crossed.”

Transparency

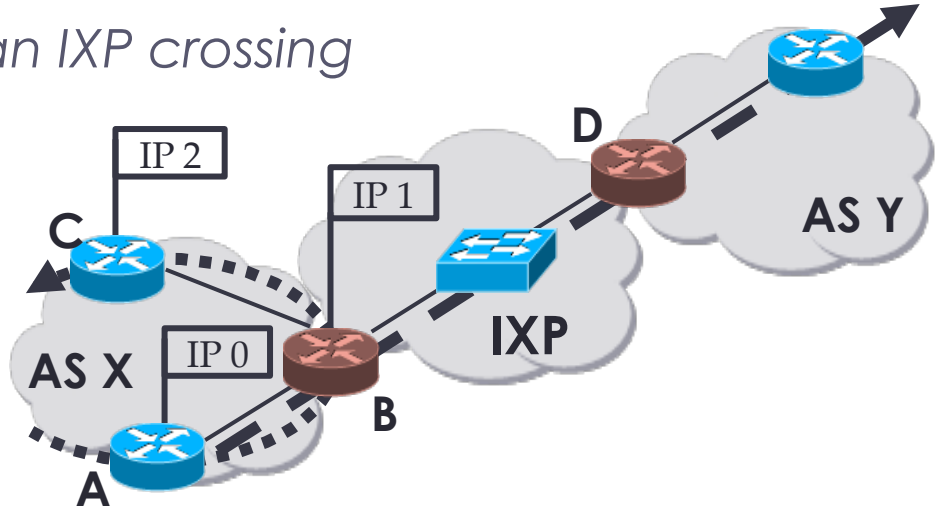
Evolution

End-to-End paths Troubleshooting

Challenge

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**

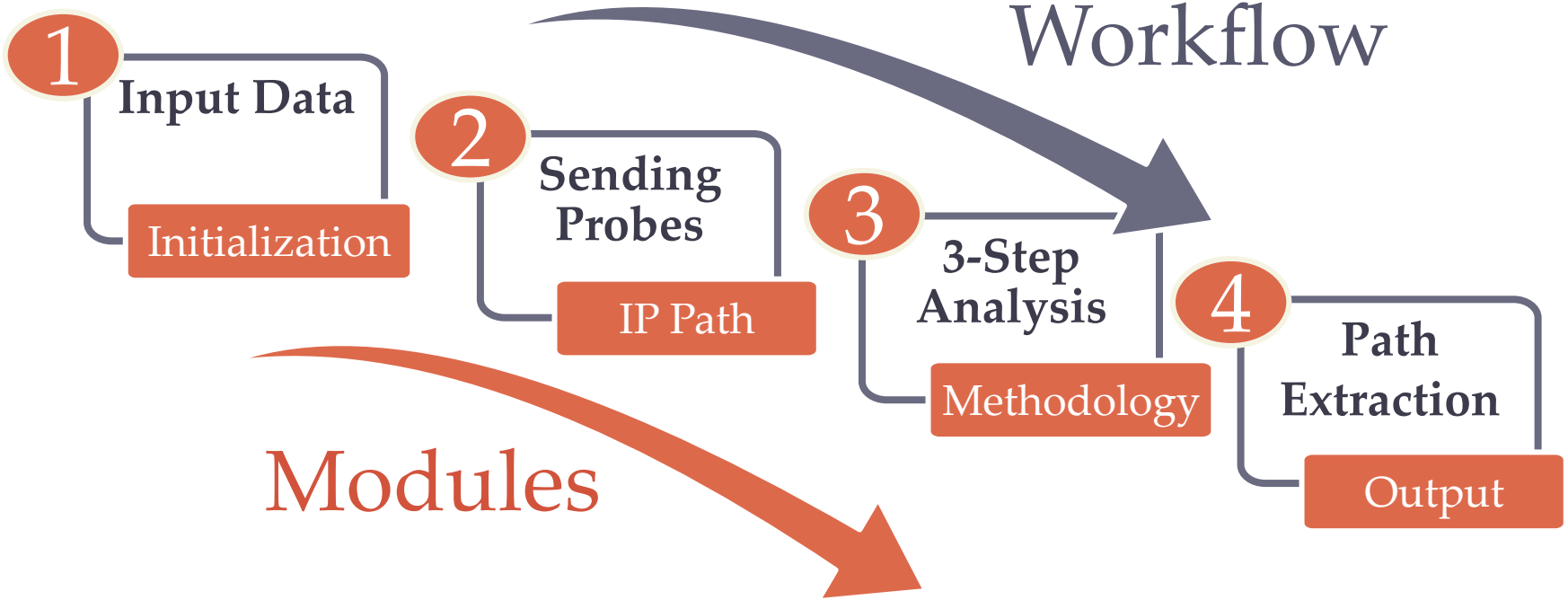
[1] G. Nomikos, X. Dimitropoulos. "***traIXroute: Detecting IXPs in traceroute paths***". In Proceedings of the Passive and Active Measurements Conference (PAM'16) 31 March - 1 April 2016, Heraklion, Greece.

tra*IX*route: 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

[1] Remote Peering Jedi tool (www.inspire.edu.gr/rp)

Modular Design & Workflow



Initialization - Input Data

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*

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**
- and the **92.1%** from **PCH**

1 2 3 4

IP Path Reception

- We send the probe to a certain destination using:
  - **Traceroute**

    or

  - **Scamper** [1]

[1] https://www.caida.org/tools/measurement/scamper

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**

1 2 3 4

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**

1 2 3 4

Methodology – Step 1

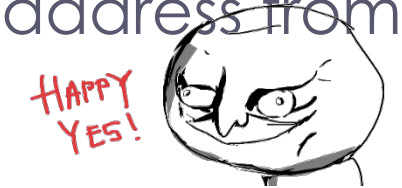
- We apply a sliding window of size **2** or **3** IP addresses.

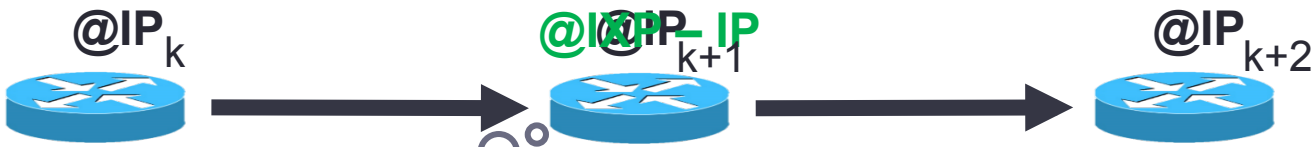
IP path

1 2 3 4

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

op window

1 2 3 4

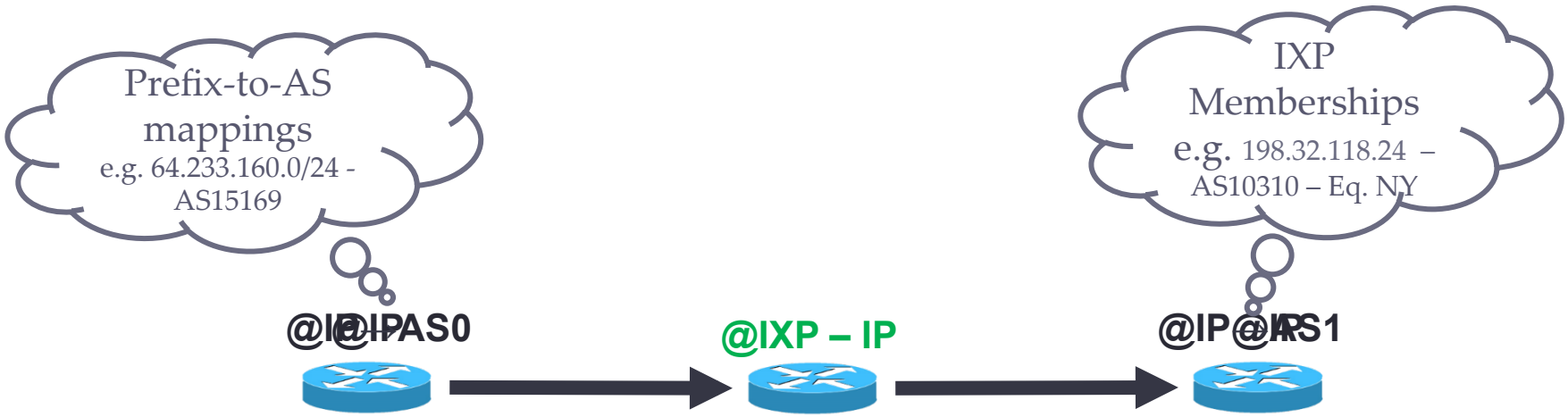
Methodology Overview

- 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**

1 2 3 4

Methodology – Step 2

- Are the adjacent ASes members of the IXP?



1 2 3 4

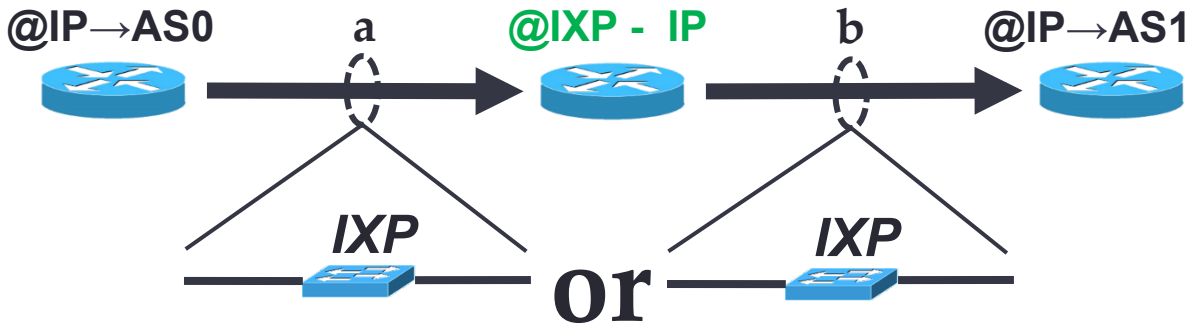
Methodology Overview

- 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**

1 2 3 4

Methodology – Step 3

- Is the IXP link crossed before or after the IXP IP address?
  - Check when sufficient information about the ASes is available.



IXP Detection Rules

We propose **strong** and **weak** evidence rules

Strong : $IP \xrightarrow{bgp} AS$ (AS) $\neq$ $IXP - IP \xrightarrow{inf} AS$ (AS) $=$ $IP \xrightarrow{bgp} AS$ (AS) ⇒ **a**

Weak : $IP \xrightarrow{bgp} AS$ (AS) $IXP - IP \xrightarrow{prf} IXP$ (IXP) $IP \xrightarrow{bgp} AS$ (AS) ⇒ **a**

traIXroute Output (1)

```
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.
```

1 2 3 4

traIXroute Output (2)

```
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).
1)   AS*          10.184.252.246 (10.184.252.246) 0.749 ms
2)   AS48293      union-tel.192.166.ru (94.228.192.166) 0.957 ms
3)   AS48293      union-tel.207.126.ru (94.228.207.126) 1.006 ms
4)   (DE-CIX)->AS20485    frt01.transtelecom.net (80.81.194.117) 41.73 ms
5)   AS20485      vvk06.transtelecom.net (217.150.59.102) 154.193 ms
6)   AS20485      IP-Kozitskiy-gw.transtelecom.net (217.150.59.101) 149.7 ms
7)   AS196949     Po1-20g.c65.vss.core.vl.podryad.tv (109.110.48.146) 150.148 ms
8)   AS196949     109.110.48.141 (109.110.48.141) 153.489 ms
9)   AS196949     atlas-probe1.ripe.noc.podryad.tv (109.110.48.142) 155.008 ms
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)
```

1 2 3 4

Use Case: IXPs in traceroute paths

- Methodology
  - **31.8 million** probed paths collected from the CAIDA's Ark measurement infrastructure*
  - **16** IXP detection rules

*Data collected on January, 20th 2015

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

Conclusions

European Research Council
netvolution.eu

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

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

Thank You!!!

gnomikos @ ics.forth.gr