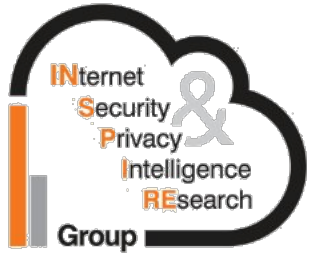


traIXroute

Detecting IXPs in traceroute paths

inspire.edu.gr/traIXroute

George Nomikos

gnomikos @ ics.forth.gr

Prof. Xenofontas Dimitropoulos

UoC / FORTH – INSPIRE Group

"… if and where an IXP was crossed."

Transparency

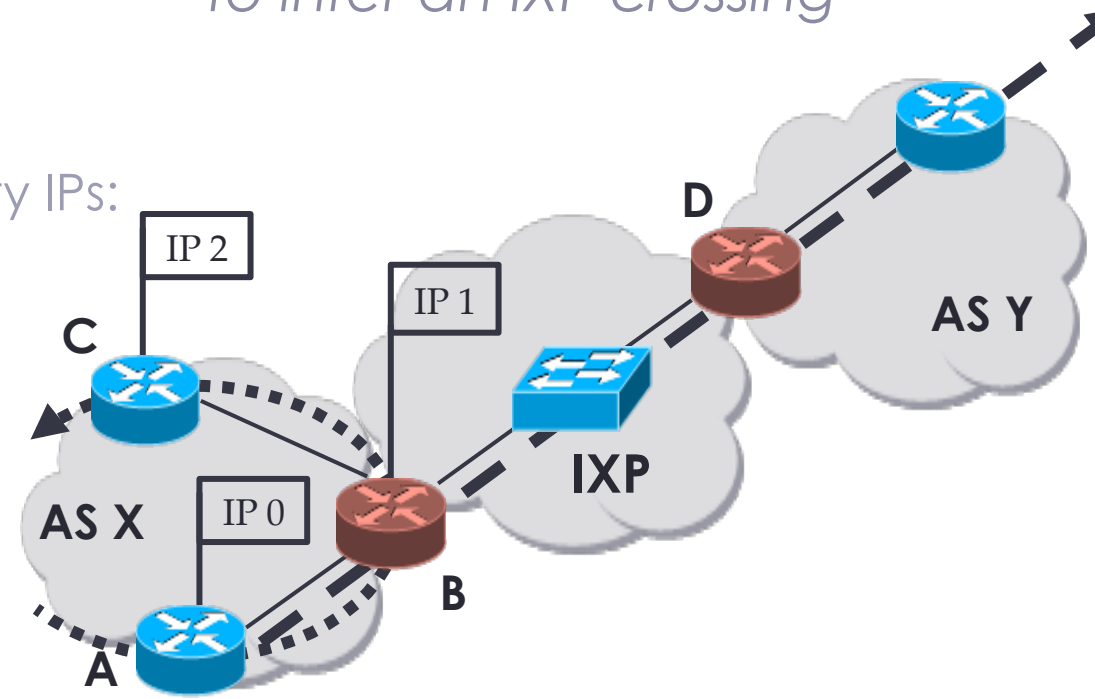
Evolution

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:

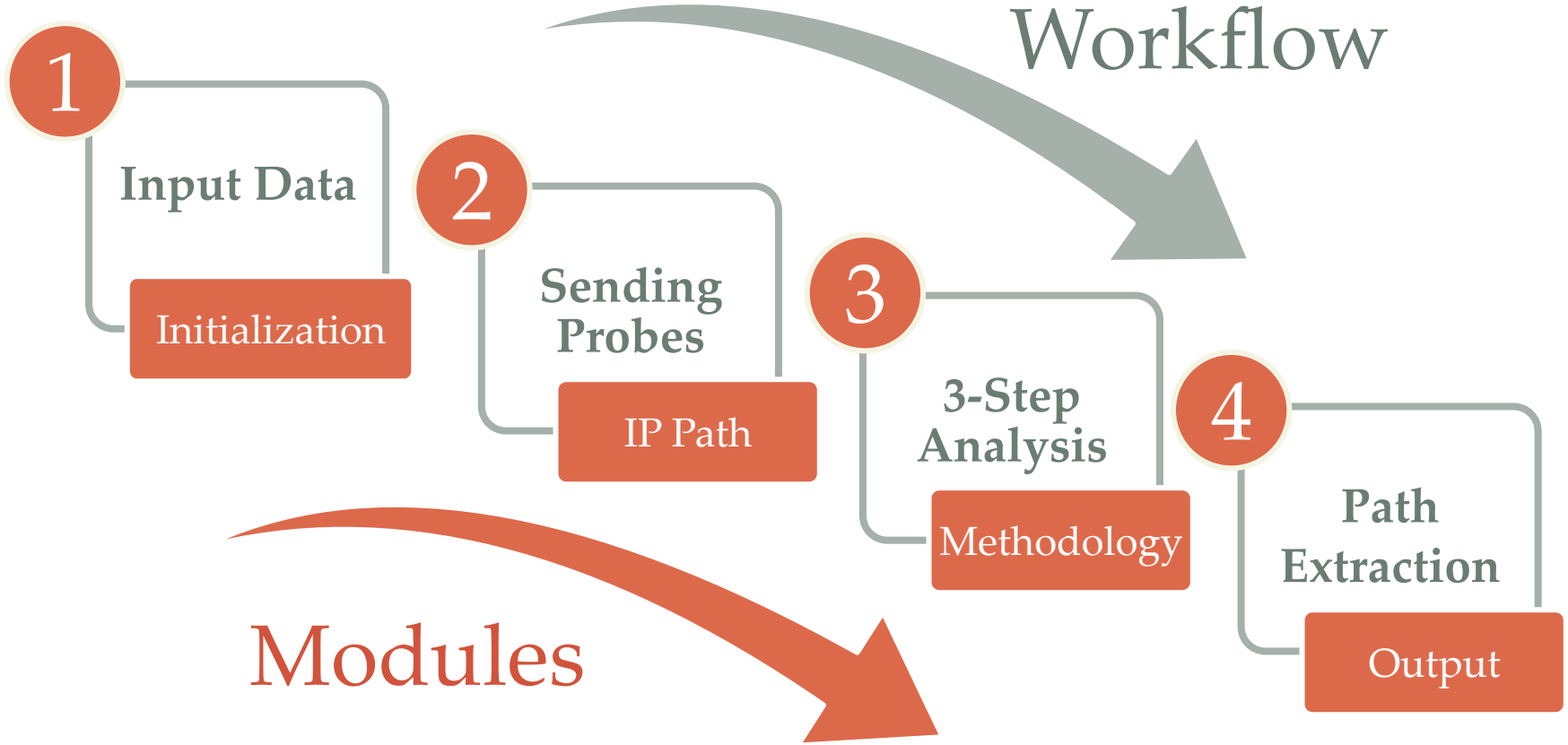


2. The available IXP prefix data may be: **a)** inaccurate or **b)** could be used in other subnets

Why traIXroute?

- ✓ A general-purpose (Python 3) tool to detect IXP hops
- ✓ Exploits only easily accessible IXP data
- ✓ Overcomes some of the existing shortcomings
- ✓ Detects IXPs in **~10** seconds
- ✓ Modular design and customization

Modular Design & Workflow



Initialization - Input Data

Provided by:

1. **IXP Memberships**
   - *e.g. Equinix New York – AS10310 – 198.32.118.24*
2. **IXP Subnets**
   - *e.g. 198.32.118.0/24 – Equinix New York*

PeeringDB & Packet Clearing House

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

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
  - Traceroute
  - Scamper

Methodology Overview

The IXP identification mechanism proceeds as follows:

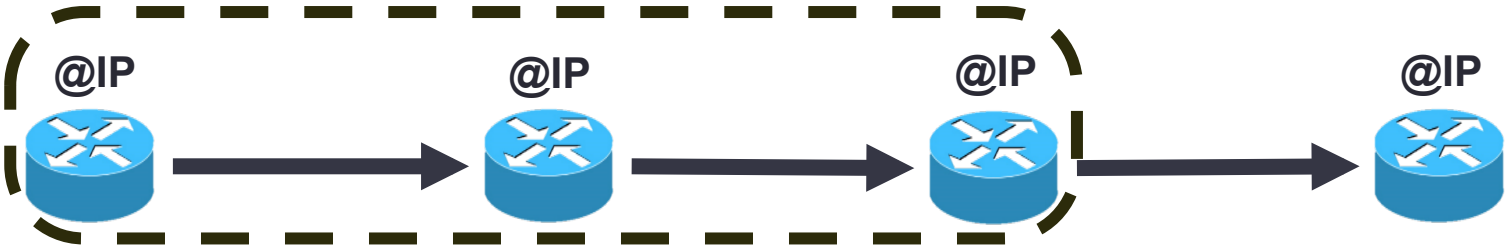
- Step 1: Detect IXP IPs in traceroute paths based on **IXP Membership data** and/or **prefixes**
- Step 2: **Check the IXP membership** of the ASes adjacent to the observed IXP address(es)
- Step 3: Identify the **IXP crossing link**

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 1

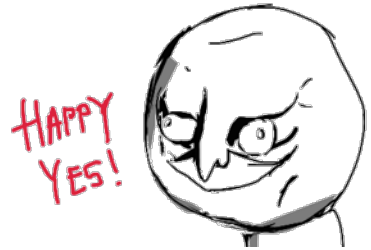
- We apply a sliding window of size **2** or **3** IP addresses.



IP path

Methodology – Step 1

- Does the IP address in hop_{k+1} match an exact BGP router IP address from an IXP subnet?



^ hop window

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

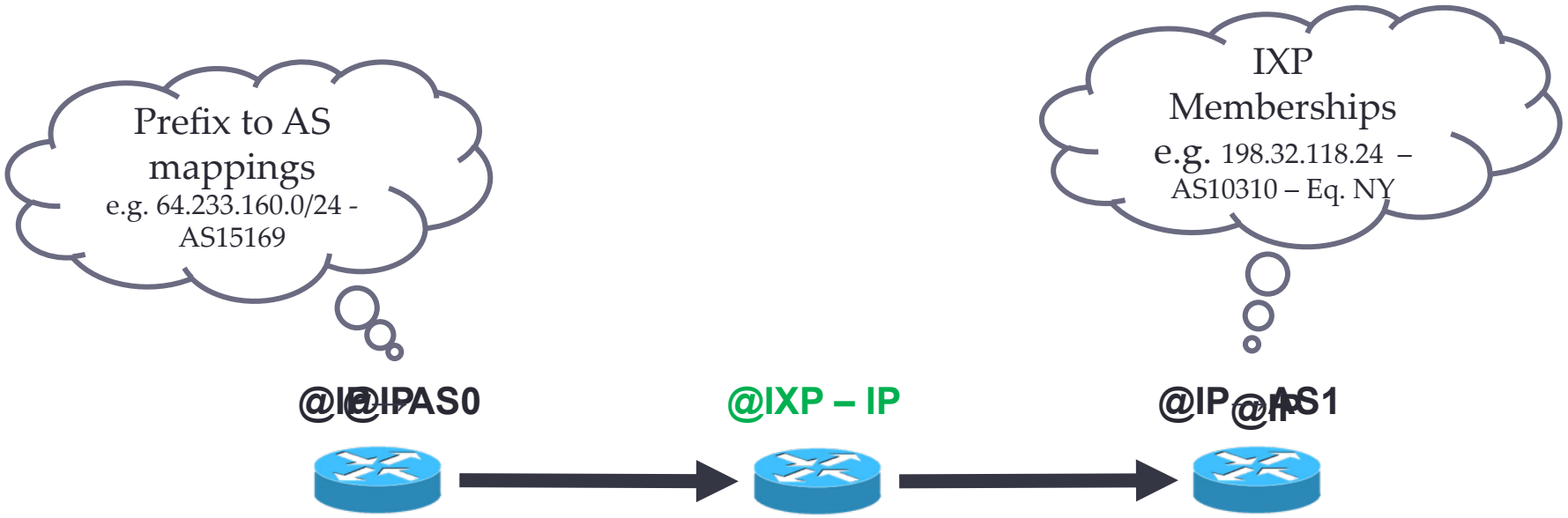
1 2 3 4

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**

Methodology – Step 2

- Are the adjacent ASes members of the IXP?

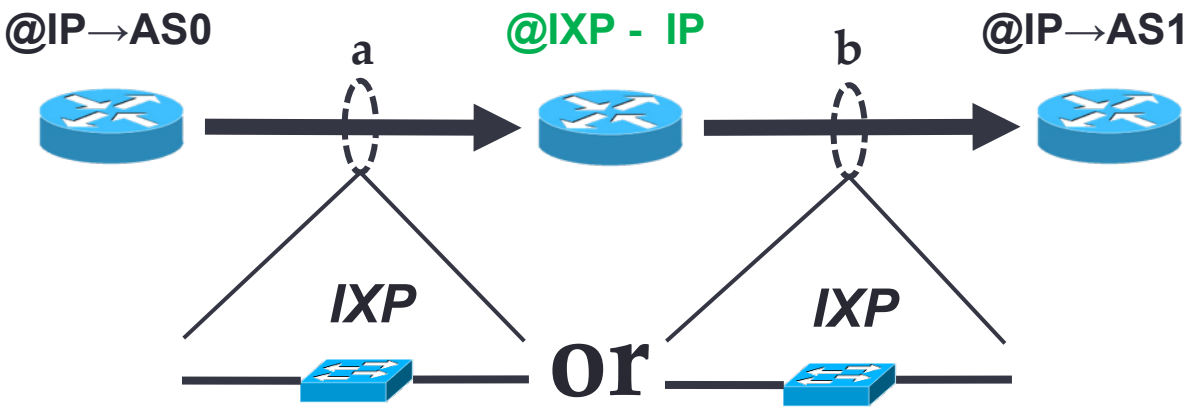


Methodology Overview

- 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**

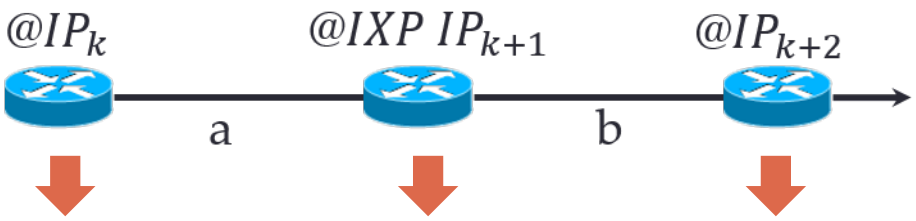
Methodology – Step 3

- Is the IXP link crossed before or after the IXP IP address?
  - 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 $IP \xrightarrow{bgp} AS$ (AS) ⇒ **a**

traIXroute Output

```
traIXrouting to inspire.edu.gr
1)    AS*          192.168.1.1 (192.168.1.1) 3.080 ms
2)    AS1241       bbras-llu-her-01L500.forthnet.gr (213.16.246.X) 30.750 ms
3)    AS1241       213.16.247.X (213.16.247.X) 30.683 ms
4)    AS1241       te0-4-0-11.core-kln-13.forthnet.gr 0 41.178 ms
5)    AS1241       distr-kln-02Be2.forthnet.gr (213.16.247.X) 37.480 ms
6)    AS1241       core-kln-12Be3.forthnet.gr (213.16.247.X) 40.440 ms
7)    GR-IX->AS5408          grnet.gr-ix.gr (176.126.38.1) 39.864 ms
8)    AS5408       forth-her-4.eier.access-link.grnet.gr (62.217.98.X) 45.995 ms
9)    AS*          (*) -
10)   AS*          (*) -
11)   AS*          (*) -
12)   AS*          (*) -
13)   AS*          (*) -
IXP Hops:
Rule: 1 --- 6) 213.16.247.17 (AS1241) <--- GR-IX ---> 7) 176.126.38.1 (AS5408)
```

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

- tra*lX*route, a useful tool to identify IXP hops in IP paths
  - **~20%** of the traceroute paths crosses one IXP
  - **Download** from: inspire.edu.gr/tralXroute
  - G. Nomikos et al., *tralXroute: Detecting IXPs in traceroute paths*, **PAM** 2016
- Ongoing & future work:
  - Used in the IXP Jedi RIPE Atlas Hackathon project
  - IPv6 support
  - Further validation

Thank You!!!

gnomikos @ ics.forth.gr

What is out there?

- Multiple research works infer **IXP peerings** via traceroute paths **using mechanisms** like:

  e.g. – *Majority-selection process*
  – *DNS naming association*
  – *Targeted tracerouting*
  – *BGP vs. Traceroute AS paths*

- Traditionally, they exploit data like:
  - BGP table dumps
  - DNS names
  - IXP prefixes
  - BGP policies

traIXroute

```
$ sudo python3 traIXroute.py -i inspire.edu.gr
```

traIXroute Output

```
sudo python3 traIXroute.py -i inspire.edu.gr -asn -rule -s
```

```
Imported 8 IXP Detection Rules from rules.txt.
Imported 16 Reserved Subnets.
Extracted 0 IXP IPs from additional_info.txt.
Extracted 1 IXP Subnets from additional_info.txt.
Extracted 14040 IXP IPs from PDB.
Extracted 9984 IXP IPs from PCH.
Extracted 377 IXP Subnets from PDB.
Extracted 368 IXP Subnets from PCH.
Extracted 14449 not dirty IXP IPs after merging PDB, PCH and
additional_info.txt.
Extracted 1516 dirty IXP IPs after merging PDB, PCH and
additional_info.txt.
Extracted 587 IXP Subnets after merging PDB, PCH and additional_info.txt.
```

Methodology Evaluation

- **31.8 million** trace probes collected from the CAIDA's Ark measurement infrastructure*
  - A total set of **107** monitors distributed around the globe
  - Monitors are split into **three teams** of similar size
  - Monitors are configured with the **scamper** tool
- **16** IXP detection rules were applied

*Data collected on January, 20th 2015

Evaluation Results

- We met two consecutive IXP IP addresses in the same traceroute path.
  - How often? …2.09%
- This happens due to:
  - Inefficient routing due to the BGP path selection process
  - Outbound and inbound responses from the BGP routers into the IXP fabric

Evaluation Results

- How often we ***infer/hit*** an IXP crossing observing one IXP IP?

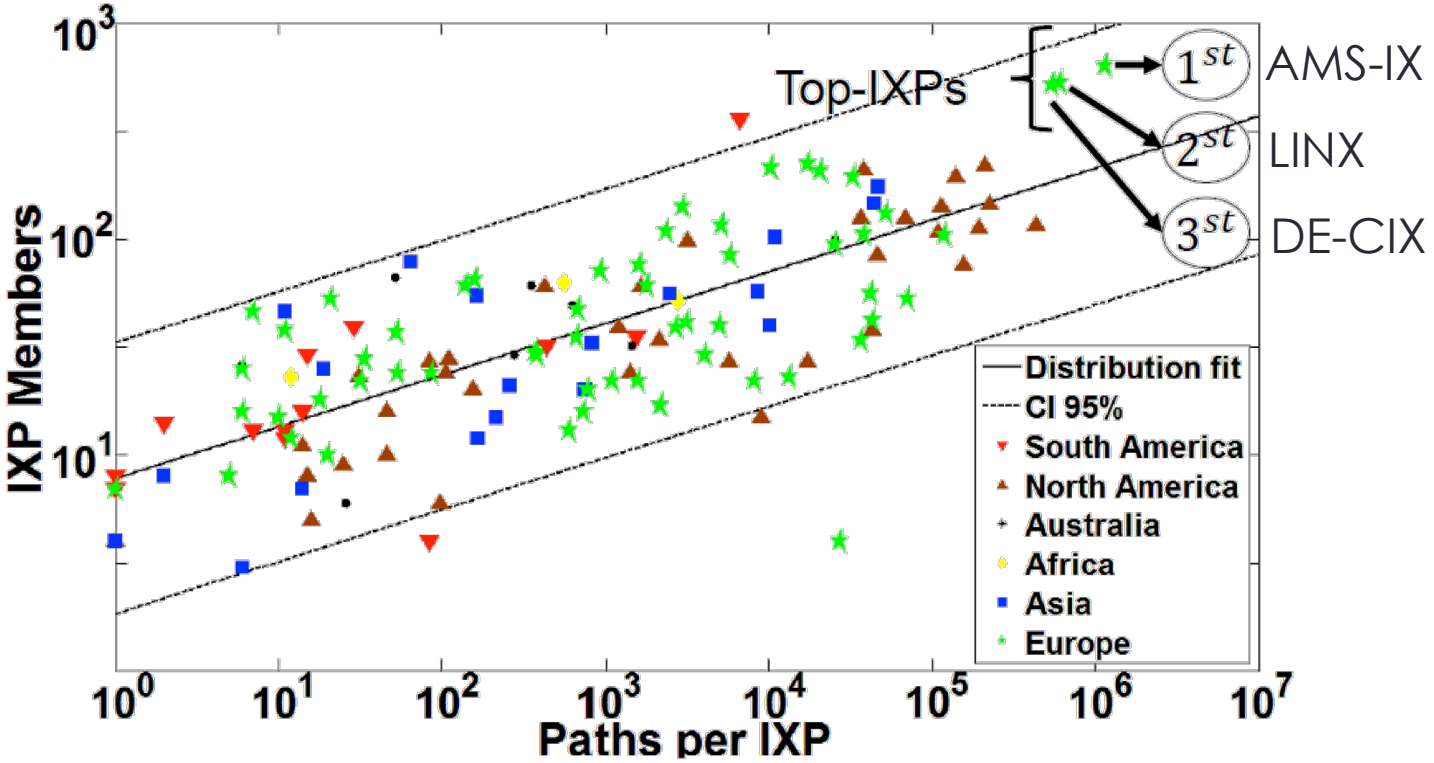
Hit Rate

3 Strong Rules ⇨ 76,86 % of the paths in which at least one IXP IP exists.

⇩

- IXP crossings are detected in most cases
- PDB and PCH rich enough to match most IXP addresses

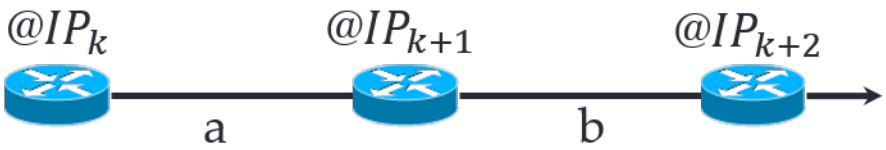
Members *vs.* Paths



Number of IXP members *vs.* Number of paths per IXP
(Correlation coefficient ρ = 0.8)

IXP Detection rules

16 distinct heuristics with **1** subsequent IXP IPs were applied



		$IP \xrightarrow{bgp} AS$	$IXP - IP \xrightarrow{inf} AS$ / $IXP - IP \xrightarrow{prf} IXP$	$IP \xrightarrow{bgp} AS$	Assessment
Strong	1	AS ≠	AS =	AS	a
	2	AS ≠	AS ≠	AS	a
	3	AS ≠	AS ≠	AS	a or b
Weak	4	AS	IXP	AS	a
	5	AS	IXP	AS	b
	6	AS ≠	AS =	AS	b
	7	AS ≠	AS ≠	AS	a or b

Evaluation Results

	$IP \xrightarrow{bgp} AS$		$IXP-IP \xrightarrow{inf} AS$ $IXP-IP \xrightarrow{prf} \text{IXP}$		$IP \xrightarrow{bgp} AS$	Hit Rate
1	AS	≠	AS	=	AS	65.57 %
2	AS	≠	AS	≠	AS	8.79 %
3	AS	≠	AS	≠	AS	2.5 %
4	AS		IXP		AS	7.7 %
5	AS		IXP		AS	5.55 %
6	AS	≠	AS	=	AS	4.56 %
7	AS	≠	AS	≠	AS	1.21 %
					SUM:	95.88%