Learning Regexes to Extract Network Names from Hostnames

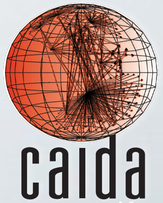
Matthew Luckie

University of Waikato

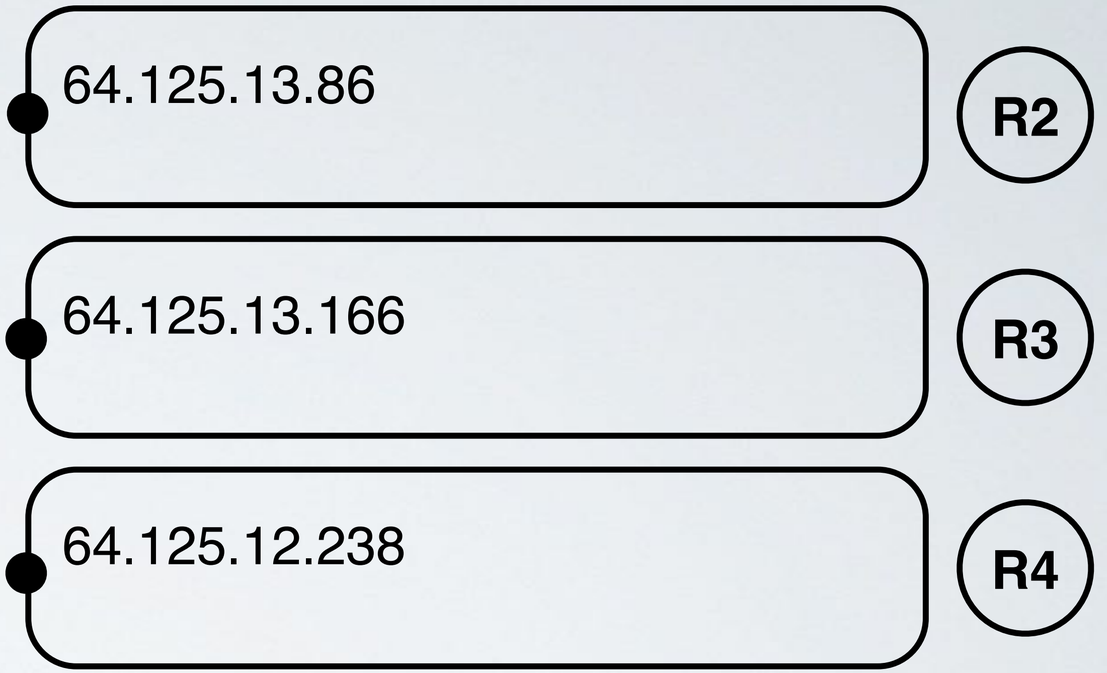
Alexander Marder
Bradley Huffaker
k claffy

CAIDA, UC San Diego

AINTEC 2021

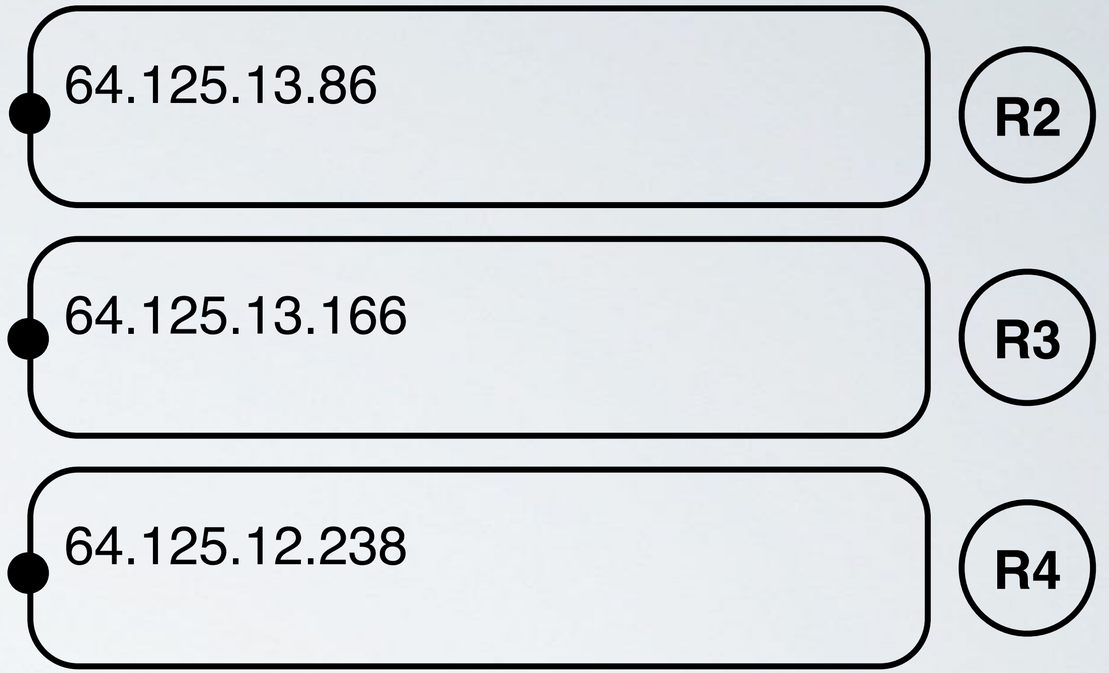


Which network operates these routers?



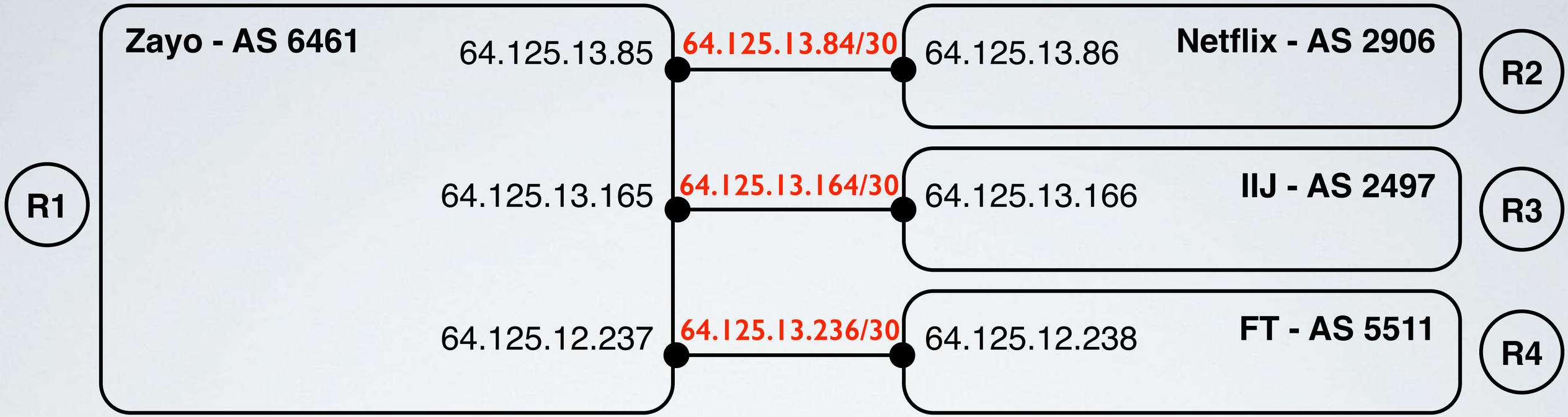
Which network operates these routers?

Historic approach: use origin Autonomous System Number (ASN) of corresponding longest matching prefix observed in BGP



64.125.0.0/16 announced by AS 6461 (Zayo)

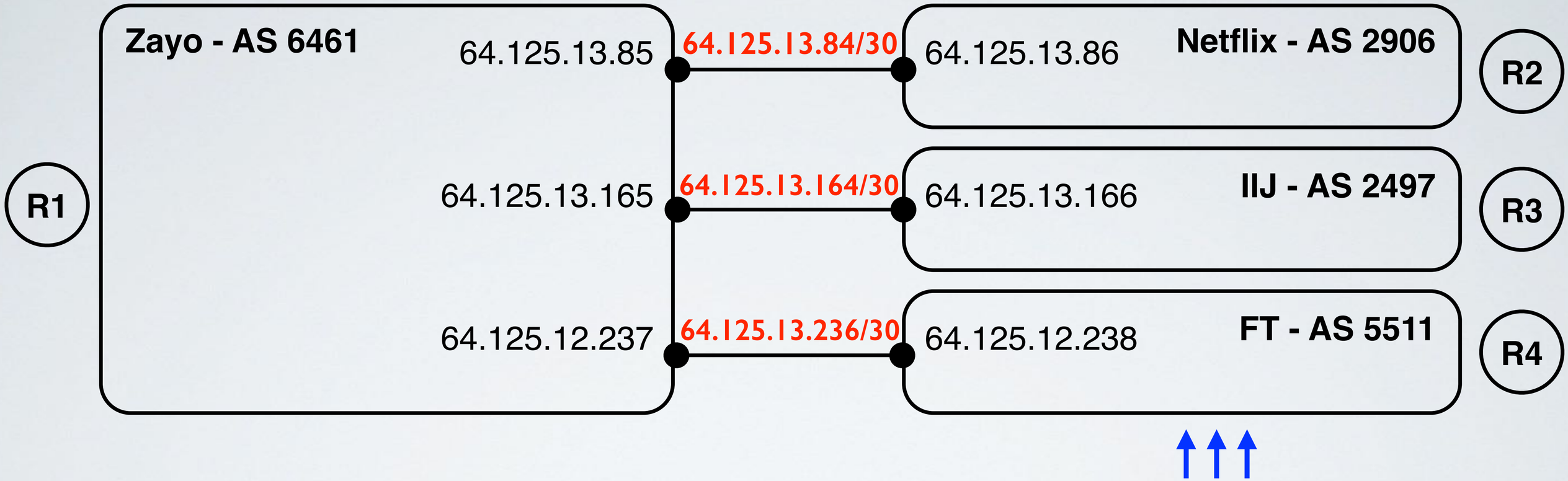
Why is this hard?



Real world: Zayo provides addresses to routers operated by their neighbors so that they can connect using the Internet protocols.

64.125.0.0/16 announced by AS 6461 (Zayo)

Why is this hard?



It is challenging to infer the operator of **AS border routers** as the router could have IP addresses provided by their neighbors.

64.125.0.0/16 announced by AS 6461 (Zayo)

A growing body of work depends on getting router ownership inference correct

Experiences Deploying Multi-Vantage-Point Domain Validation at Let's Encrypt

The Art of Detecting Forwarding Detours

Investigating the Causes of Congestion on the African IXP substrate

Inferring Persistent Interdomain Congestion

Latency Imbalance Among Internet Load-Balanced Paths: A Cloud-Centric View

Revealing the Load-balancing Behavior of YouTube Traffic on Interdomain Links

A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet

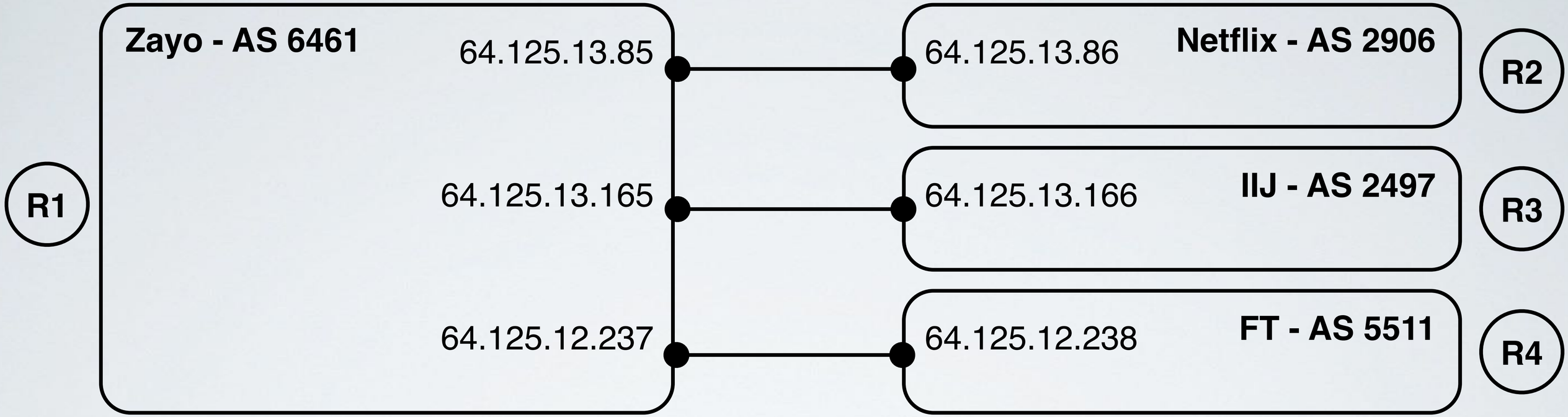
More examples cited in the paper

Unintended consequences: Effects of submarine cable deployment on Internet routing

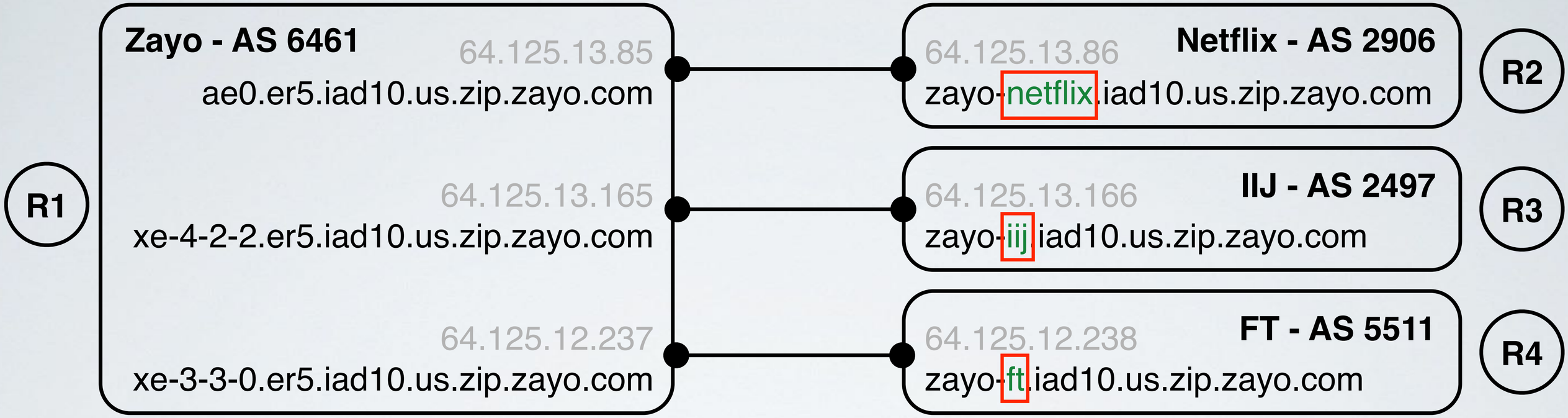
Router Ownership Inference Techniques: Limited Validation

Towards an Accurate AS-Level Traceroute Tool	SIGCOMM 2003
Scalable and Accurate Identification of AS-Level Forwarding Paths	INFOCOM 2004
Toward Topology Dualism: Improving the Accuracy of AS Annotations for Routers	**PAM 2010 (RTAA)**
bdrmap: Inference of Borders Between IP Networks	IMC 2016
MAP-IT: Multipass Accurate Passive Inferences from Traceroute	IMC 2016
Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale	**IMC 2018 (bdrmapIT)**

Our approach: use information in hostnames

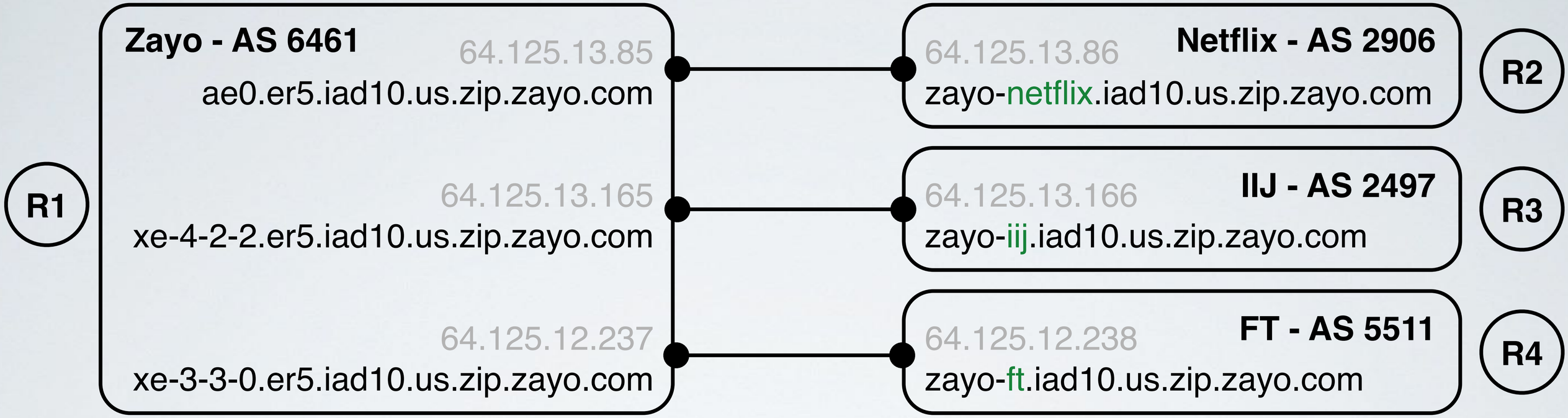


Our approach: use information in hostnames



Some operators embed information in hostnames because it helps them (and others) debug (and understand) their networks

Our approach: use information in hostnames



Zayo Router: er5.iad.us

^zayo-([a-z\d]+)\.[a-z]{3}\d+\.[a-z]{2}\.zip\.zayo\.com$

Zayo embeds the name of the neighbor network in a ***predictable format***, i.e., they use a ***naming convention***.

Holistic Hostname Orthography

- **Broader context**: automatically reason about structure in router hostnames

Holistic Hostname Orthography

ae1.mcs1.cdg11.fr.eth.zayo.com
ae61.mcs1.cdg11.fr.zip.zayo.com

ae0.mcs1.ams17.nl.eth.zayo.com
ae1.mcs1.ams17.nl.eth.zayo.com

ae15.mpr1.hnd1.jp.zip.zayo.com
ae16.mpr1.hnd1.jp.zip.zayo.com

zayo-ft.iad10.us.zip.zayo.com

zayo-iij.iad10.us.zip.zayo.com

zayo-ntt.mpr1.cdg11.fr.zip.zayo.com
as2914-gw-ptl.cw.net
ae-15.r04.parsfr01.fr.bb.gin.ntt.net
ae-2.r04.parsfr01.fr.bb.gin.ntt.net

- **Broader context**: automatically reason about structure in router hostnames
- Luckie et al. "Learning Regexes to Extract Router Names from Hostnames" IMC 2019

Holistic Hostname Orthography

- **Broader context**: automatically reason about structure in router hostnames
- Luckie et al. "Learning Regexes to Extract Router Names from Hostnames" IMC 2019
- Luckie et al. "Learning to Extract and Use ASNs in Hostnames" IMC 2020

Holistic Hostname Orthography

ae1.mcs1.cdg11.fr.eth.zayo.com
ae61.mcs1.cdg11.fr.zip.zayo.com

ae0.mcs1.ams17.nl.eth.zayo.com
ae1.mcs1.ams17.nl.eth.zayo.com

ae15.mpr1.hnd1.jp.zip.zayo.com
ae16.mpr1.hnd1.jp.zip.zayo.com

zayo-ft.iad10.us.zip.zayo.com

zayo-iij.iad10.us.zip.zayo.com

zayo-ntt.mpr1.cdg11.fr.zip.zayo.com
as2914-gw-ptl.cw.net
ae-15.r04.parsfr01.fr.bb.gin.ntt.net
ae-2.r04.parsfr01.fr.bb.gin.ntt.net

- **Broader context**: automatically reason about structure in router hostnames
- Luckie et al. "Learning Regexes to Extract Router Names from Hostnames" IMC 2019
- Luckie et al. "Learning to Extract and Use ASNs in Hostnames" IMC 2020
- Luckie et al. "Learning to Extract Geographic Information from Internet Router Hostnames" CoNEXT 2021

Holistic Hostname Orthography

ae1.mcs1.cdg11.fr.eth.zayo.com
ae61.mcs1.cdg11.fr.zip.zayo.com

ae0.mcs1.ams17.nl.eth.zayo.com
ae1.mcs1.ams17.nl.eth.zayo.com

ae15.mpr1.hnd1.jp.zip.zayo.com
ae16.mpr1.hnd1.jp.zip.zayo.com

zayo-**ft**.iad10.us.zip.zayo.com

zayo-**iij**.iad10.us.zip.zayo.com

zayo-**ntt**.mpr1.cdg11.fr.zip.zayo.com
as2914-gw-ptl.cw.net
ae-15.r04.parsfr01.fr.bb.gin.ntt.net
ae-2.r04.parsfr01.fr.bb.gin.ntt.net

- **Broader context**: automatically reason about structure in router hostnames
- Luckie et al. "Learning Regexes to Extract Router Names from Hostnames" IMC 2019
- Luckie et al. "Learning to Extract and Use ASNs in Hostnames" IMC 2020
- Luckie et al. "Learning to Extract Geographic Information from Internet Router Hostnames" CoNEXT 2021
- Luckie et al. "Learning Regexes to Extract Network Names from Hostnames" AINTEC 2021

Contributions of this work

- **We design and implement a method that automatically**
  - **learns regexes** that extract ***network names*** from hostnames,
  - **learns dictionary** that maps ***network names*** to their ***ASN***
- **We publicly release**
  - **the source code** implementation as part of ***Hoiho***,
    (Hoiho: Holistic Orthography of Internet Hostname Observations)
  - **the inferred naming conventions**
    - https://www.caida.org/tools/measurement/scamper/
    - https://www.caida.org/publications/papers/2021/hoiho-asnames/

Hoiho: Yellow-eyed penguin

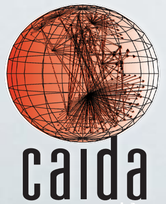
Image: Brent Beaven
Department of Conservation (New Zealand)

CAIDA Internet Topology Data Kit (ITDK)

Heavily curated router-level topology dataset published roughly twice a year

- IPv4 Routers, with aliases inferred by MIDAR and Mercator
- IPv6 Routers, with aliases inferred by Speedtrap
- Links between routers
- Router geolocation
- Router ownership (which AS operates each router)
  - RouterToAsAssignment (July 2010 - February 2017)
  - bdrmapIT (August 2017 - March 2021)
- DNS hostnames
- 19 ITDK datasets between July 2010 to March 2021

Hoiho Input Data



CAIDA Internet Topology Data Kit (ITDK)

- IPv4 and IPv6 Routers
- Router ownership (which AS operates each router)
- DNS hostnames

Intuition: operators in different networks tend to use the same string to identify a neighbor

Hostname	ASN	Naming Network
zayo-**netflix**.iad10.us.zip.zayo.com	2906	Zayo
netflix.sgix.sg	2906	Singapore IX
netflix-gw.customer.alter.net	2906	Verizon
netflix1.fra.ecix.net	2906	Megaport ECIX
netflix1-lacp-100g.hkix.net	2906	Hong Kong IX
netflix-inc.ear2.sanjose1.level3.net	2906	Level3
nflx1.ix.fl-ix.net	2906	Community IX Florida

The corresponding routers are operated by AS 2906: Netflix.

Most operators use the same string "netflix".

Some variation: abbreviation of "netflix" as "nflx".

Note: operators have different conventions for how they encode information in hostnames.

Approach

1. Automatically Build Seed Name Dictionary
2. Automatically Build Seed Regular Expressions
3. Automatically Refine Name Dictionary
4. Automatically Refine Regular Expressions

Approach

1. **Automatically Build Seed Name Dictionary**
2. Automatically Build Seed Regular Expressions
3. Automatically Refine Name Dictionary
4. Automatically Refine Regular Expressions

Name	ASN
level3	3356
netflix	2906
iij	2497
zayo	6461
gig	64496

Build a seed dictionary that **maps strings to ASNs**, with the observation that **multiple** different suffixes (**operators**) **using the same string** provides grounds to infer a **seed name**

Approach

1. Automatically Build Seed Name Dictionary
2. **Automatically Build Seed Regular Expressions**
3. Automatically Refine Name Dictionary
4. Automatically Refine Regular Expressions

Name	ASN
level3	3356
netflix	2906
iij	2497
zayo	6461
gig	64496

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$

Build seed regexes that extract **seed network names**, to allow phase three to identify and remove spurious entries in the seed name dictionary

Approach

1. Automatically Build Seed Name Dictionary
2. Automatically Build Seed Regular Expressions
3. **Automatically Refine Name Dictionary**
4. Automatically Refine Regular Expressions

Name	ASN
level3	3356
netflix	2906
iij	2497
zayo	6461

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$

Use seed regexes, which are likely to only extract network names, **to rebuild the dictionary from scratch**

Approach

1. Automatically Build Seed Name Dictionary
2. Automatically Build Seed Regular Expressions
3. Automatically Refine Name Dictionary
4. **Automatically Refine Regular Expressions**

Name	ASN
level3	3356
netflix	2906
iij	2497
zayo	6461

Rebuild regexes with refined dictionary, investing time to **build specific regexes** that **reflect operator intent**

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$

^zayo-([a-z\d]+)\.[a-z]{3}\d+\.[a-z]{2}\.zip\.zayo\.com$

Phase 1: Build Seed Dictionary

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
netflix-gw.skt.cw.net	2906
level3-gw.dus.cw.net	3356
netflix-ic-324205-hls-b2.c.telia.net	2906
level3-ic-347052-hls-b1.c.telia.net	3356

Identify strings in hostnames separated by punctuation.

Phase 1: Build Seed Dictionary

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
netflix-gw.skt.cw.net	2906
level3-gw.dus.cw.net	3356
netflix-ic-324205-hls-b2.c.telia.net	2906
level3-ic-347052-hls-b1.c.telia.net	3356

Identify strings in hostnames separated by punctuation.

netflix **level3** **zip**

cdg11 **hls** **c** **iad10** **gw** **b2**

us **skt** **fr** **dus** **b1** **ic** **zayo**

Phase 1: Build Seed Dictionary

2906: zayo.com
2906: cw.net
2906: telia.net

Identify strings in hostnames separated by punctuation.

Annotate strings with ASN:suffix tags

Phase 1: Build Seed Dictionary

netflix

2906: zayo.com
2906: cw.net
2906: telia.net

level3

3356: zayo.com
3356: cw.net
3356: telia.net

Identify strings in hostnames separated by punctuation.

Annotate strings with ASN:suffix tags

Phase 1: Build Seed Dictionary

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
netflix-gw.skt.cw.net	2906
level3-gw.dus.cw.net	3356
netflix-ic-324205-hls-b2.c.telia.net	2906
level3-ic-347052-hls-b1.c.telia.net	3356

Identify strings in hostnames separated by punctuation.

Annotate strings with ASN:suffix tags

netflix

2906: zayo.com
2906: cw.net
2906: telia.net

level3

3356: zayo.com
3356: cw.net
3356: telia.net

zip

2906: zayo.com
3356: zayo.com

Phase 1: Build Seed Dictionary

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
netflix-gw.skt.cw.net	2906
level3-gw.dus.cw.net	3356
netflix-ic-324205-hls-b2.c.telia.net	2906
level3-ic-347052-hls-b1.c.telia.net	3356

Add
netflix: 2906
level3: 3356
to seed dictionary

3 different suffixes label routers operated by those ASNs with those strings

netflix
2906: zayo.com
2906: cw.net
2906: telia.net

level3
3356: zayo.com
3356: cw.net
3356: telia.net

zip
2906: zayo.com
3356: zayo.com

Phase 1: Build Seed Dictionary

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
netflix-gw.skt.cw.net	2906
level3-gw.dus.cw.net	3356
netflix-ic-324205-hls-b2.c.telia.net	2906
level3-ic-347052-hls-b1.c.telia.net	3356

Do not add **zip** to seed dictionary: no clear signal that zip is correlated with any ASN

netflix

2906: zayo.com
2906: cw.net
2906: telia.net

level3

3356: zayo.com
3356: cw.net
3356: telia.net

zip

2906: zayo.com
3356: zayo.com

Phase 2: Build Seed Regexes for each suffix

Hostname	ASN	
zayo-netflix.iad10.us.zip.zayo.com	2906	(A)
zayo-level3.cdg11.fr.zip.zayo.com	3356	(B)
zayo-tata.ams1.nl.zip.zayo.com	6453	(C)
zayo-sprint.er2.ord7.us.zip.zayo.com	1239	(D)
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453	(E)
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956	(F)

Seed Name Dictionary
netflix:2906 level3:3356 tata:6453
sprint:1239 telefonica:12956

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$ (RE1)
True Positives: 3 of 6 — hostnames A, B, C

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$
True Positives: 3 of 6 — hostnames D, E, F (RE2)

^[^-]+-([^\.]+)\..+\.zayo\.com$ (RE3)
True Positives: 6 of 6 — hostnames A, B, C, D, E, F

Identify strings in hostname with an entry in the seed dictionary congruent with the ASN mapping, then automatically build and evaluate regexes that extract those strings.

Phase 2: Build Seed Regexes for each suffix

Hostname	ASN	
zayo-netflix.iad10.us.zip.zayo.com	2906	(A)
zayo-level3.cdg11.fr.zip.zayo.com	3356	(B)
zayo-tata.ams1.nl.zip.zayo.com	6453	(C)
zayo-sprint.er2.ord7.us.zip.zayo.com	1239	(D)
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453	(E)
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956	(F)

Seed Name netflix:2906 level3:3356 tata:6453
Dictionary sprint:1239 telefonica:12956

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$ RE1
True Positives: 3 of 6 — hostnames A, B, C

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$
True Positives: 3 of 6 — hostnames D, E, F RE2

^[^-]+-([^\.]+)\..+\.zayo\.com$ RE3
True Positives: 6 of 6 — hostnames A, B, C, D, E, F

First set: zayo-<name> followed by ***three*** dot-separated components.

Zayo has slightly different formats for labelling network names in hostnames.

Phase 2: Build Seed Regexes for each suffix

Hostname	ASN	
zayo-netflix.iad10.us.zip.zayo.com	2906	(A)
zayo-level3.cdg11.fr.zip.zayo.com	3356	(B)
zayo-tata.ams1.nl.zip.zayo.com	6453	(C)
zayo-sprint.er2.ord7.us.zip.zayo.com	1239	(D)
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453	(E)
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956	(F)

Seed Name netflix:2906 level3:3356 tata:6453
Dictionary sprint:1239 telefonica:12956

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$ RE1
True Positives: 3 of 6 — hostnames A, B, C

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$
True Positives: 3 of 6 — hostnames D, E, F RE2

^[^-]+-([^\.]+)\..+\.zayo\.com$ RE3
True Positives: 6 of 6 — hostnames A, B, C, D, E, F

Second set: zayo-<name> followed by ***four*** dot-separated components.

Zayo has slightly different formats for labelling network names in hostnames.

Phase 2: Build Seed Regexes for each suffix

Hostname	ASN	
zayo-netflix.iad10.us.zip.zayo.com	2906	(A)
zayo-level3.cdg11.fr.zip.zayo.com	3356	(B)
zayo-tata.ams1.nl.zip.zayo.com	6453	(C)
zayo-sprint.er2.ord7.us.zip.zayo.com	1239	(D)
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453	(E)
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956	(F)

Seed Name netflix:2906 level3:3356 tata:6453
Dictionary sprint:1239 telefonica:12956

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$ RE1
True Positives: 3 of 6 — hostnames A, B, C

^[^-]+-([^\.]+)\.[^\.]+\.[^\.]+\.[^\.]+\.[^\.]+\.zayo\.com$
True Positives: 3 of 6 — hostnames D, E, F RE2

^[^-]+-([^\.]+)\..+\.zayo\.com$ RE3
True Positives: 6 of 6 — hostnames A, B, C, D, E, F

In practice, important thing is that the network name is found after the first dash.

Zayo has slightly different formats for labelling network names in hostnames.

Phase 3: Refine Name Dictionary

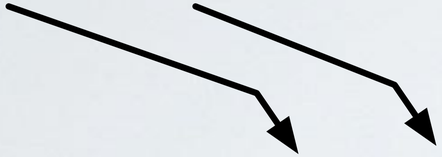
(see paper for details)

- Use seed regexes to filter out spurious dictionary entries by rebuilding dictionary from scratch
- Intuition:
  - *best regexes* per suffix will extract multiple different dictionary entries, and not spurious seed entries
  - *best regexes* across suffixes will converge on a reasonable dictionary
  - *best regexes* are likely to identify reasonable dictionary entries. therefore, we can relax criteria for including a dictionary entry, and consider abbreviations.

Name	ASN
level3	3356
netflix	2906
iij	2497
zayo	6461
gig	64496

Phase 4: Refine Regexes

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
zayo-tata.ams1.nl.zip.zayo.com	6453
zayo-sprint.er2.ord7.us.zip.zayo.com	1239
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956

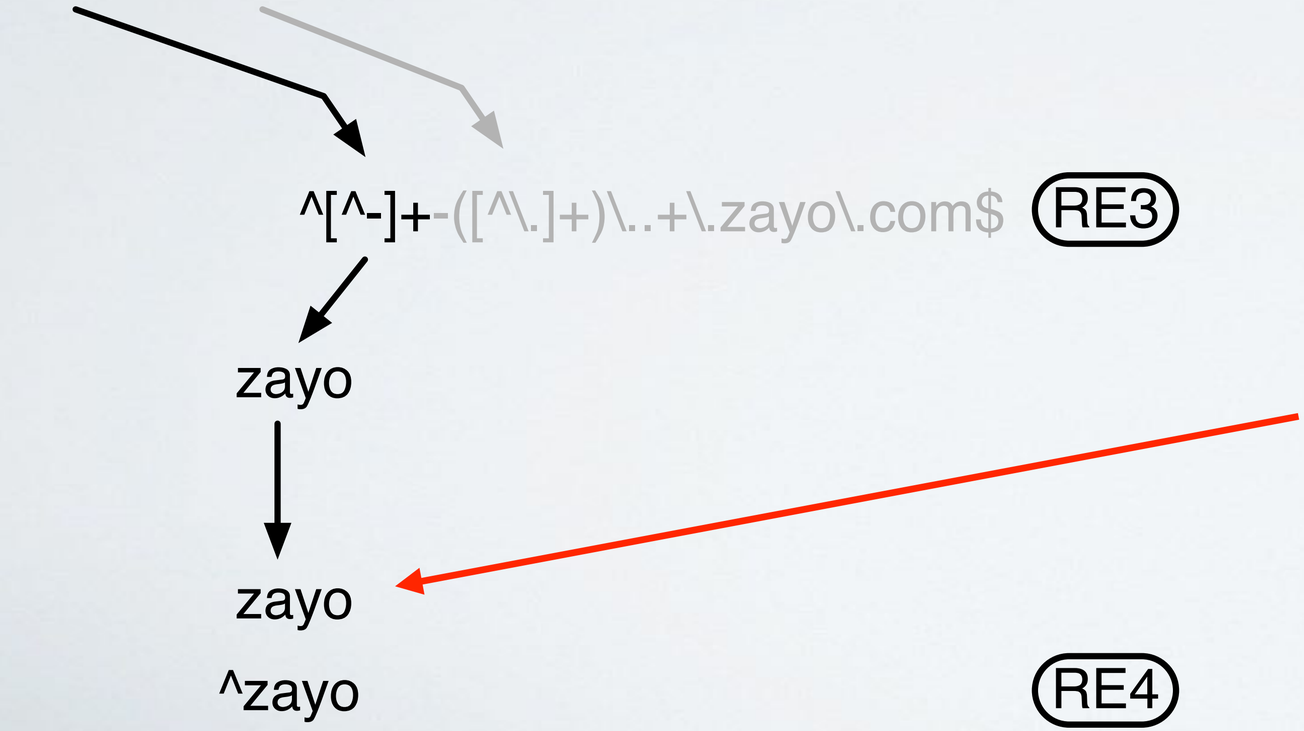


^[^-]+-([^\.]+)\..+\.zayo\.com$ RE3

Rebuild regexes with refined dictionary, investing time to build specific regexes that reflect the operator's intent

Phase 4: Refine Regexes

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
zayo-tata.ams1.nl.zip.zayo.com	6453
zayo-sprint.er2.ord7.us.zip.zayo.com	1239
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956



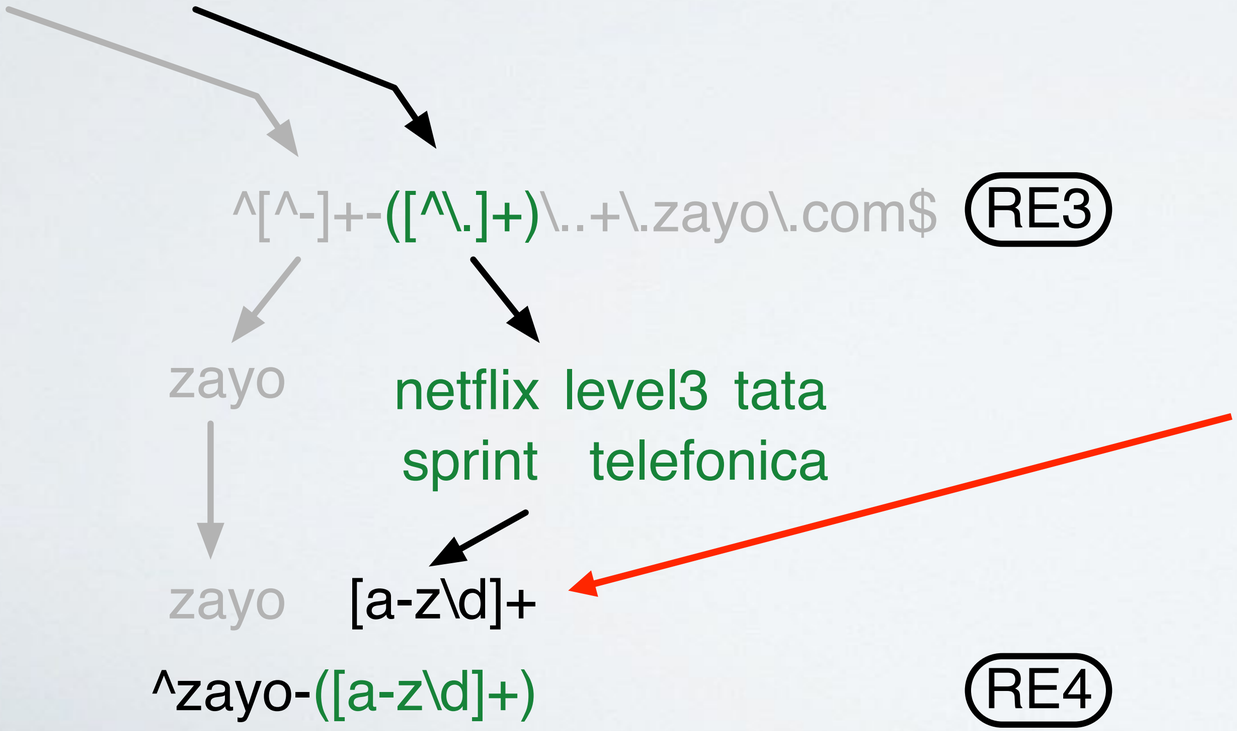
Rebuild regexes with refined dictionary, investing time to build specific regexes that reflect the operator's intent

First regex component always extracts "zayo" — our method specifies that string in the refined regex

Phase 4: Refine Regexes

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
zayo-tata.ams1.nl.zip.zayo.com	6453
zayo-sprint.er2.ord7.us.zip.zayo.com	1239
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956

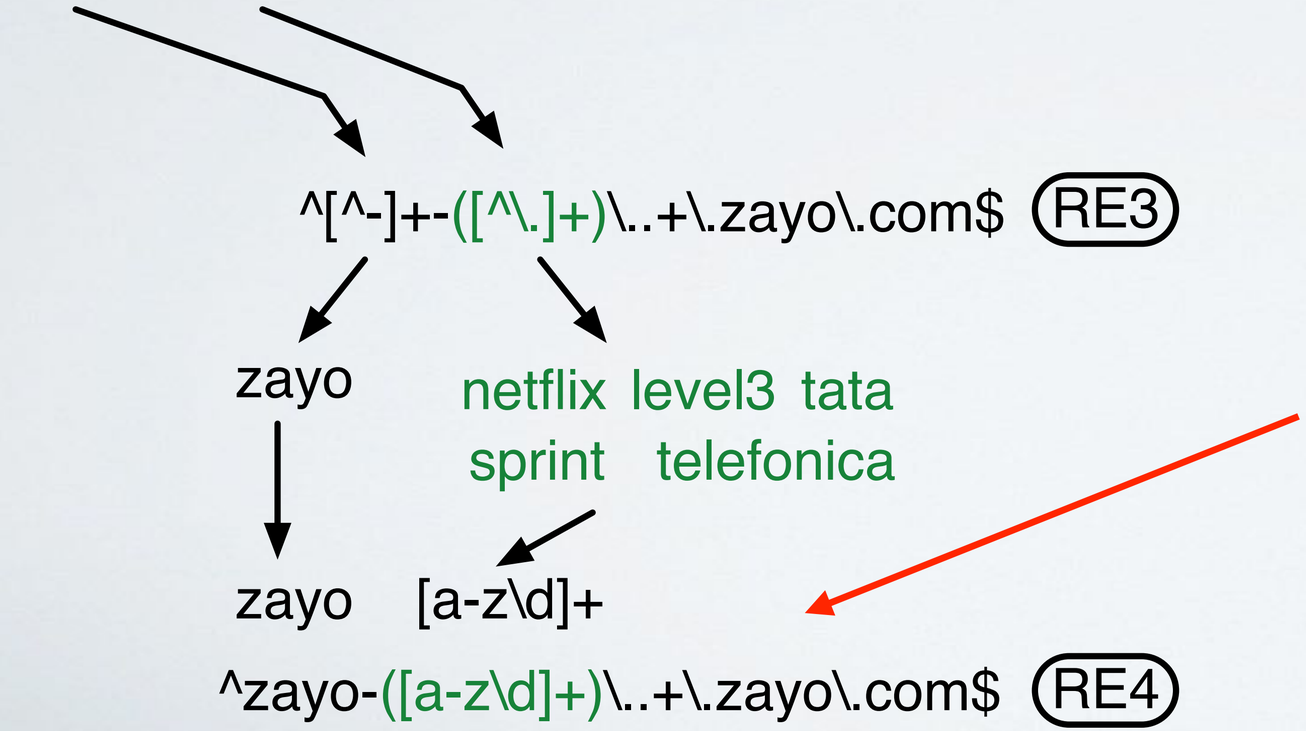
Rebuild regexes with refined dictionary, investing time to build specific regexes that reflect the operator's intent



Second regex component extracts alphanumeric strings. Our method specifies [a-z\d]+ in the refined regex

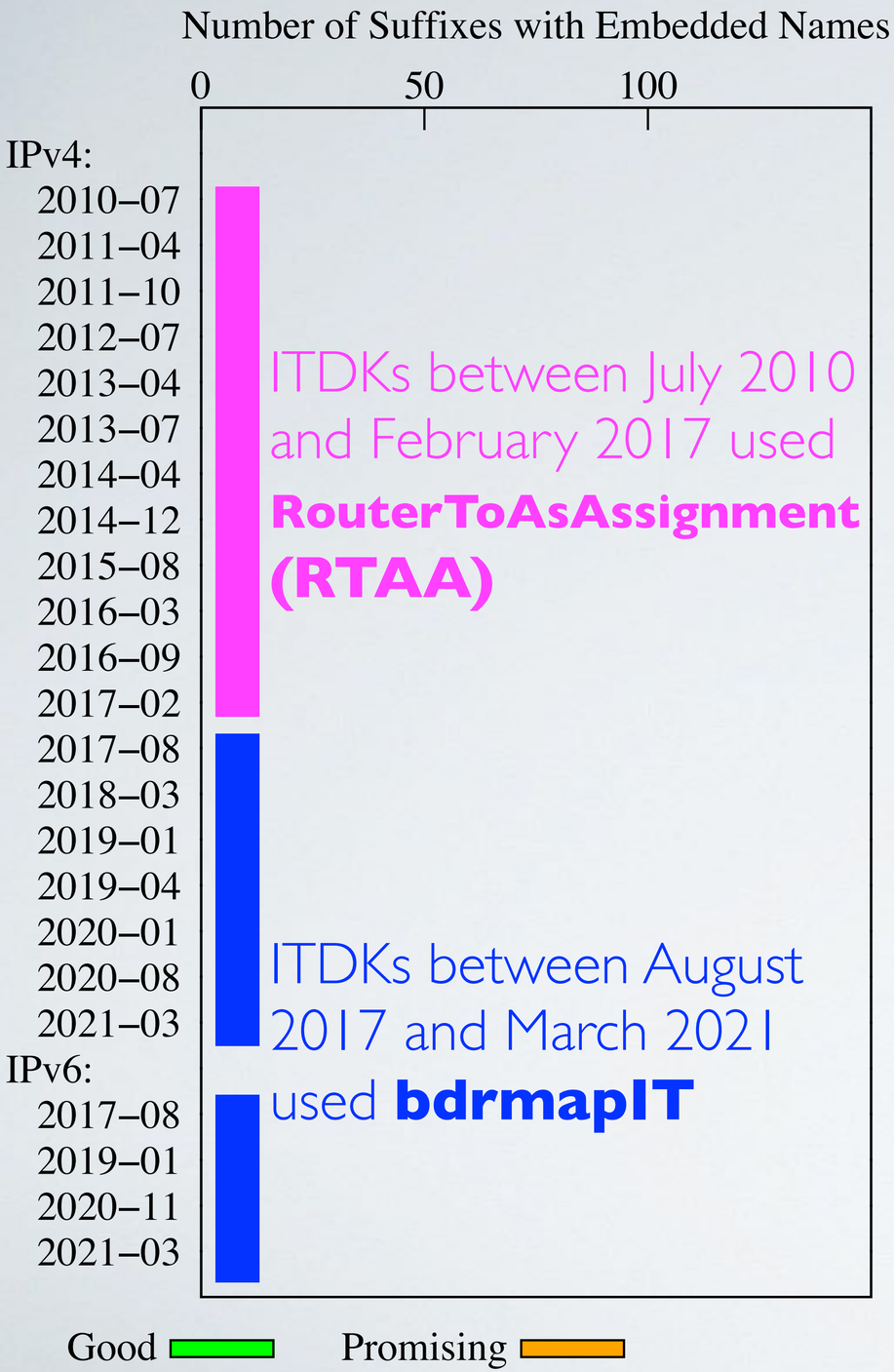
Phase 4: Refine Regexes

Hostname	ASN
zayo-netflix.iad10.us.zip.zayo.com	2906
zayo-level3.cdg11.fr.zip.zayo.com	3356
zayo-tata.ams1.nl.zip.zayo.com	6453
zayo-sprint.er2.ord7.us.zip.zayo.com	1239
zayo-tata.mpr1.fra4.de.zip.zayo.com	6453
zayo-telefonica.er2.dfw2.us.zip.zayo.com	12956



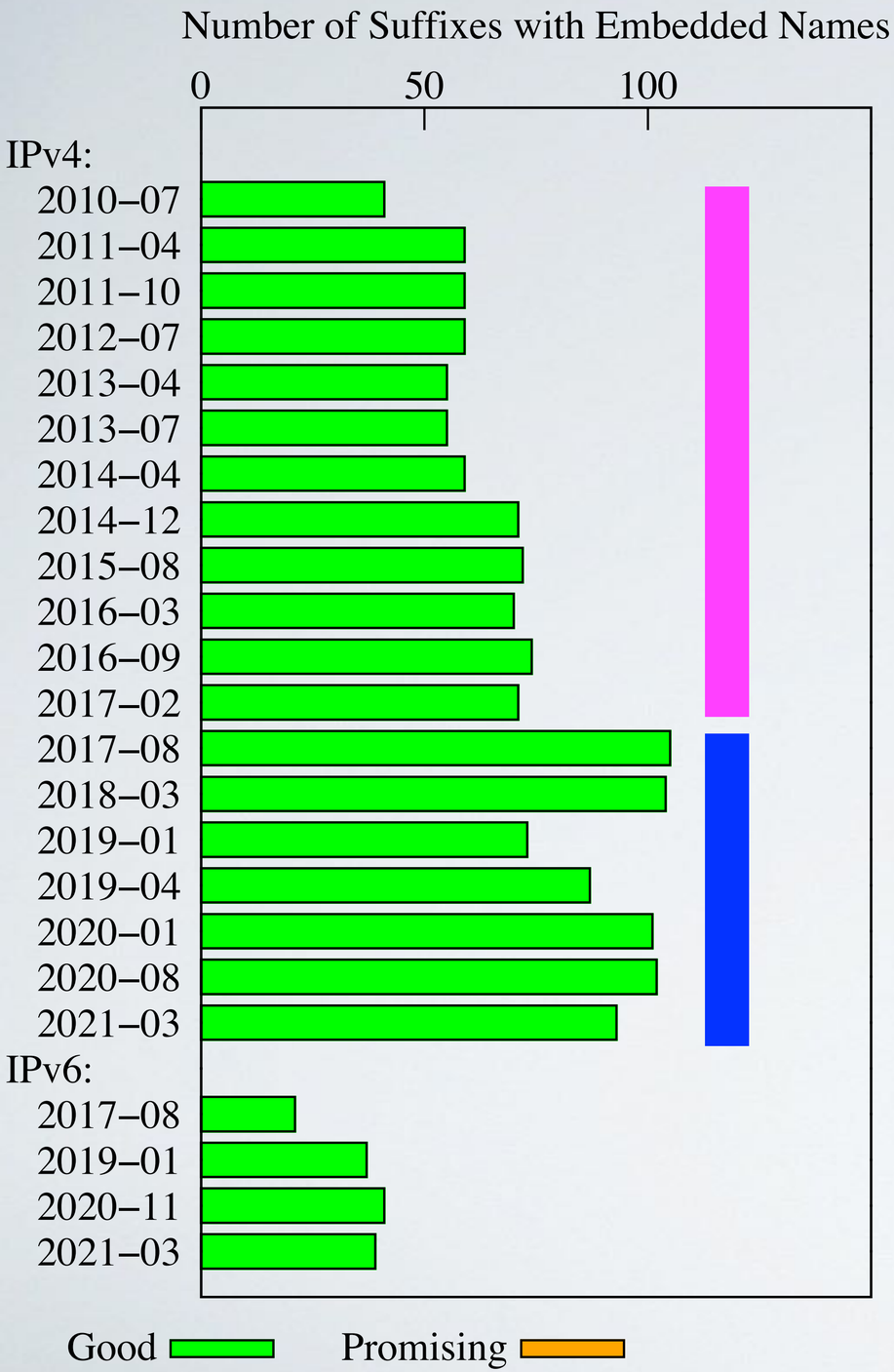
Rebuild regexes with refined dictionary, investing time to build specific regexes that reflect the operator's intent

Our method concatenates the remainder of the regex which contains no specific matching syntax



Heuristic Method Progress

Heuristic-based Router Ownership Inference Methods



Heuristic Method Progress

RTAA:

This AS names work (shown):

~62 good conventions per ITDK

Prior ASN work (not shown):

12-22 good conventions per ITDK

bdrmapIT:

This AS names work (shown):

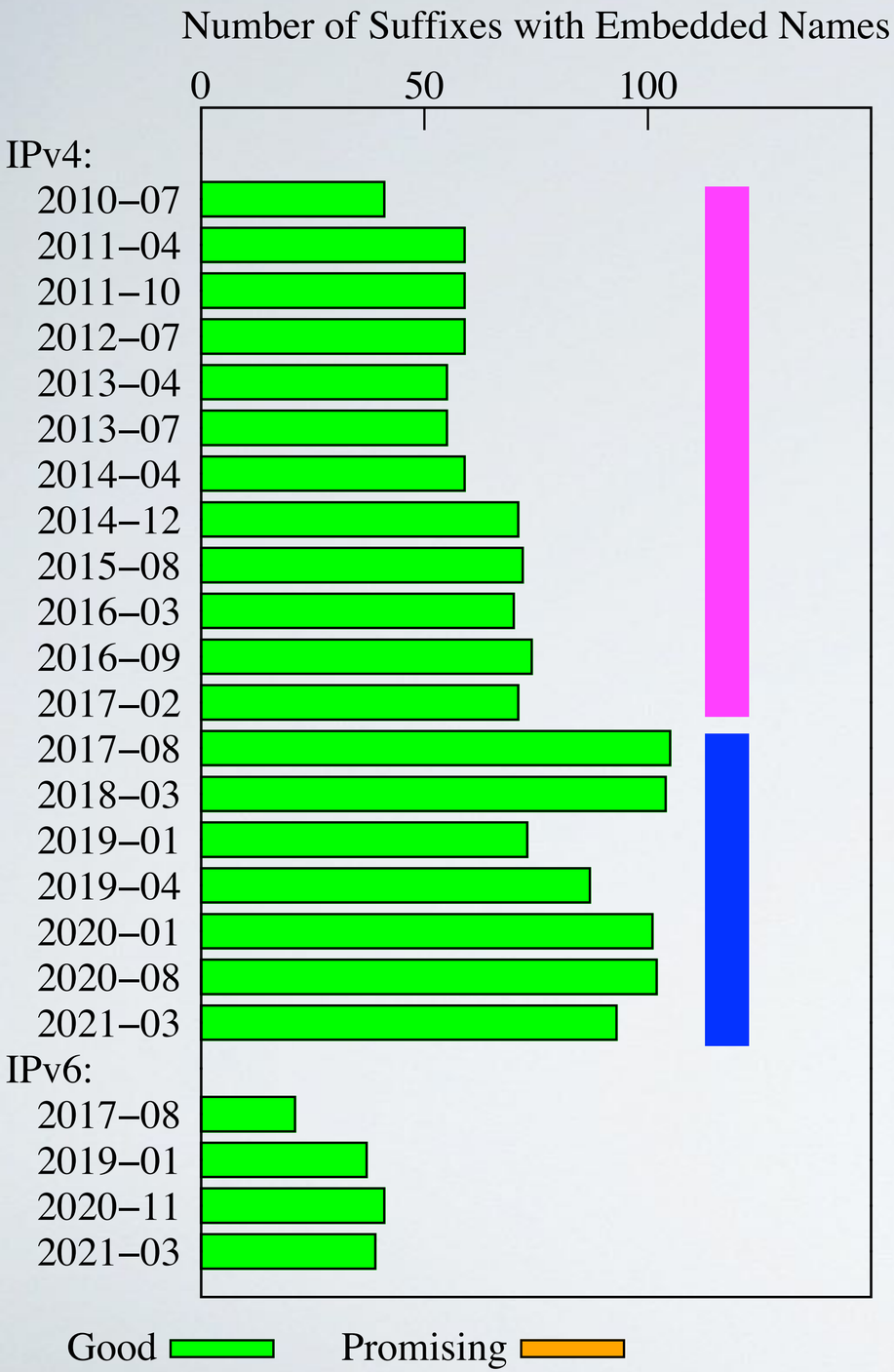
~95 good conventions per ITDK

Prior ASN work (not shown):

41-55 good conventions per ITDK

~50% more good conventions using bdrmapIT than RTAA

- **Good conventions:** PPV > 80%, >= 3 uniq network names congruent w/ training data



Heuristic Method Progress

RTAA:

This AS names work (shown):

~62 good conventions per ITDK

Prior ASN work (not shown):

12-22 good conventions per ITDK

bdrmapIT:

This AS names work (shown):

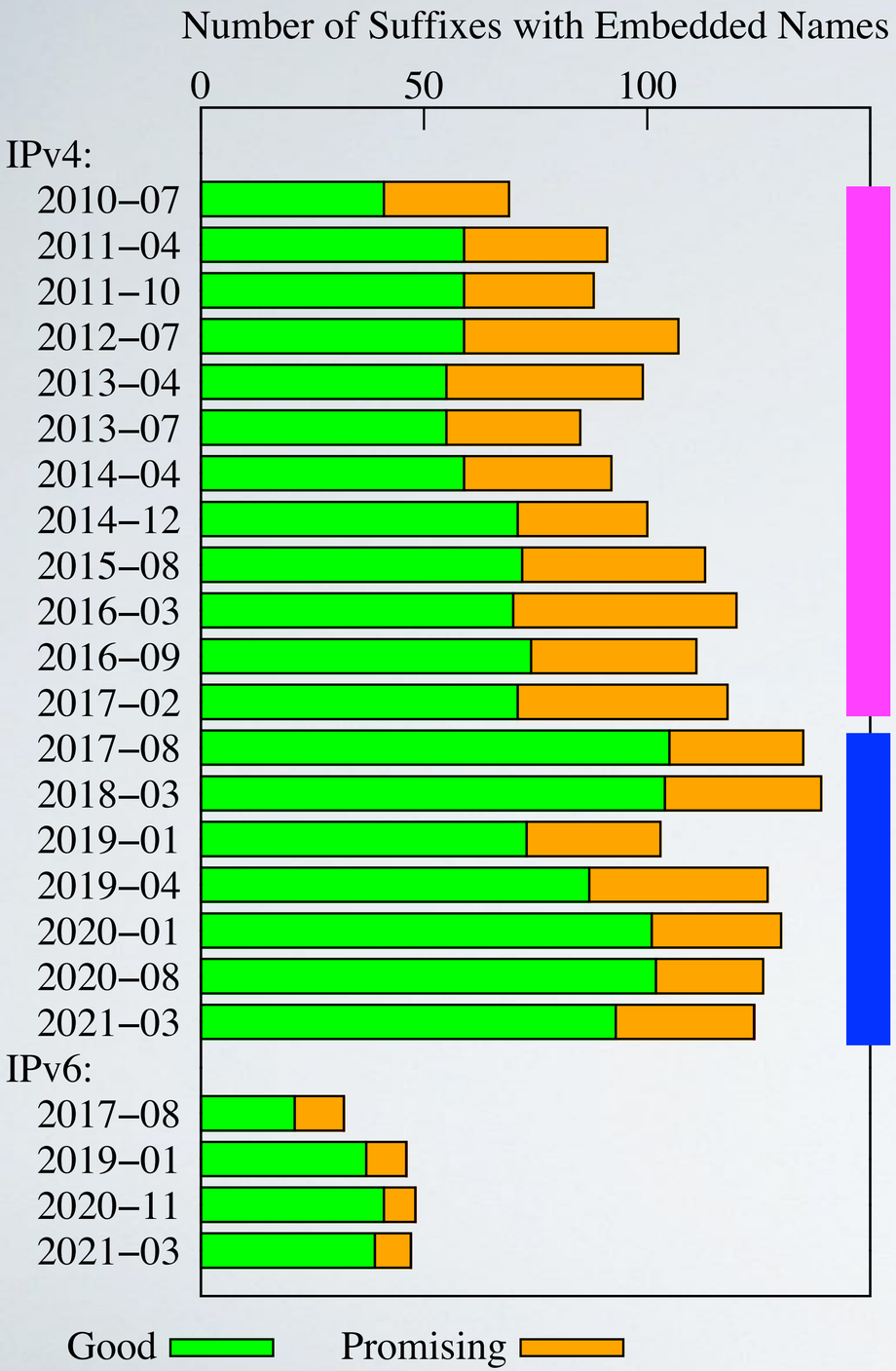
~95 good conventions per ITDK

Prior ASN work (not shown):

41-55 good conventions per ITDK

2x more good conventions using AS names than embedded ASNs

- **Good conventions:** PPV > 80%, >= 3 uniq network names congruent w/ training data



Heuristic Method Progress

RTAA:

This AS names work (shown):

~99 usable conventions per ITDK

Prior ASN work (not shown):

31-61 usable conventions per ITDK

bdrmapIT:

This AS names work (shown):

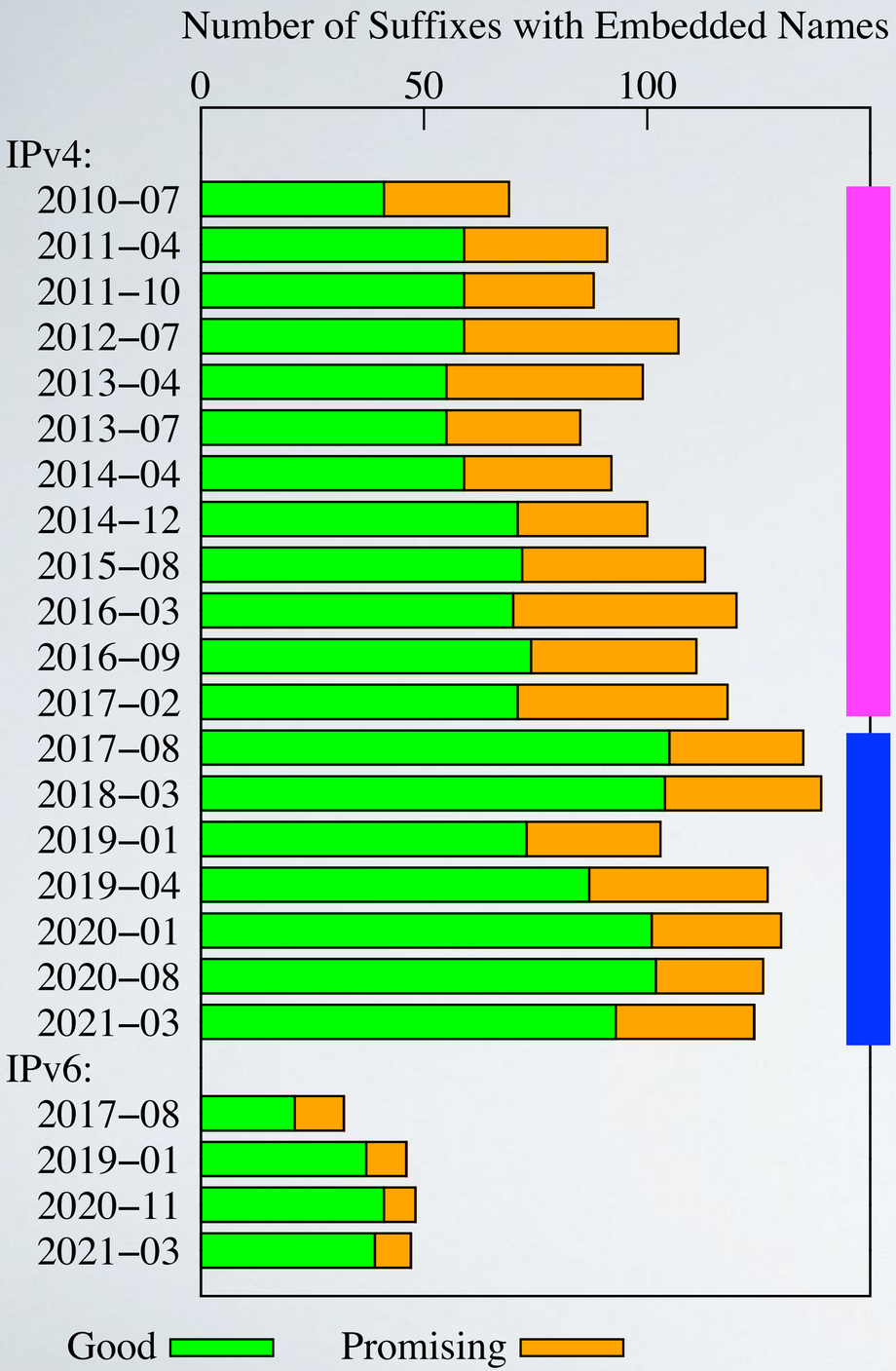
~126 good conventions per ITDK

Prior ASN work (not shown):

69-90 good conventions per ITDK

- **Promising conventions:** PPV > 50%, >=2 uniq network names congruent w/ training data

- **Good** and **Promising** are ***Usable***



Heuristic Method Progress

RTAA

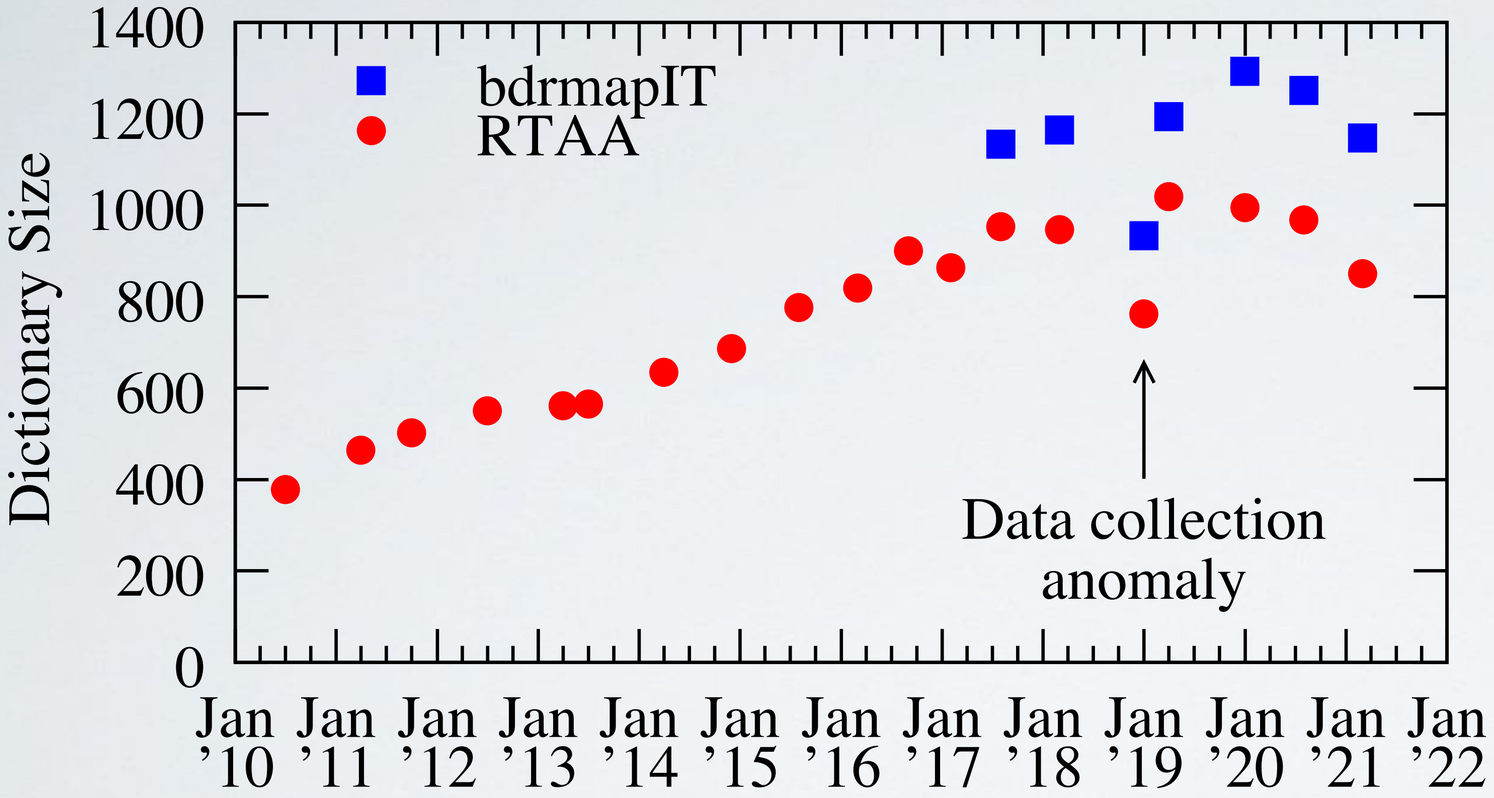
bdrmapIT

Our method inferred "usable" conventions for 308 suffixes in total.

The number of suffixes our method detected embedding an network name increased over time.

- **Promising conventions:** PPV > 50%, >=2 uniq network names congruent w/ training data
- **Good** and **Promising** are ***Usable***

Dictionary Properties



Size of dictionary increases over time as the size of the Internet increases.

Increase in network name mappings when heuristic method changed to bdrmapIT improved the ability of our method to infer network names.

Dictionary Properties

Inference of a network name is correlated with the *Customer Cone* (CC) size of the corresponding ASN.

Customer cone size	Named / Total	Coverage
0 (stub)	357 / 60535	0.6% of 84.7%
1-9	308 / 8640	3.6% of 12.1%
10-99	221 / 1882	11.7% of 2.6%
100-999	87 / 322	27.0% of 0.5%
1000-9999	31 / 43	72.1% of 0.1%
>= 10000	11 / 11	100% of 0.0%

0.6% of ASNs with zero CC

Coverage of named ASNs by customer cone size, for March 2021 ITDK.

Dictionary Properties

Inference of a network name is correlated with the *Customer Cone* (CC) size of the corresponding ASN.

Customer cone size	Named / Total	Coverage
0 (stub)	357 / 60535	0.6% of 84.7%
1-9	308 / 8640	3.6% of 12.1%
10-99	221 / 1882	11.7% of 2.6%
100-999	87 / 322	27.0% of 0.5%
1000-9999	31 / 43	72.1% of 0.1%
>= 10000	11 / 11	100% of 0.0%

6.0% of ASNs with CC >= 1

Coverage of named ASNs by customer cone size, for March 2021 ITDK.

Dictionary Properties

Inference of a network name is correlated with the *Customer Cone* (CC) size of the corresponding ASN.

Customer cone size	Named / Total	Coverage
0 (stub)	357 / 60535	0.6% of 84.7%
1-9	308 / 8640	3.6% of 12.1%
10-99	221 / 1882	11.7% of 2.6%
100-999	87 / 322	27.0% of 0.5%
1000-9999	31 / 43	72.1% of 0.1%
>= 10000	11 / 11	100% of 0.0%

77.8% of ASNs with CC >= 1000

Coverage of named ASNs by customer cone size, for March 2021 ITDK.

Validation of Dictionary

Using WHOIS (W), PeeringDB (PDB), and Manual Inspection

Source	Total	Gain	Cumulative	
W OrgName	841	841	841	75.4%
W ASName	780	108	949	85.0%
W Total	949			
PDB Name	750	27	976	87.5%
PDB Website	564	20	996	89.2%
PDB A/K/A	341	12	1,008	90.3%
PDB IRR	279	2	1,010	90.5%
PDB Total	843			
Manual	106	106	1,116	100%
Total		**1,116**		

We confirmed that **1,116 of 1,147 (97.3%)** of learned names are a valid name for the ASN

Validation of Dictionary

Using WHOIS (W), PeeringDB (PDB), and Manual Inspection

Source	Total	Gain	Cumulative	
W OrgName	841	841	841	75.4%
W ASName	780	108	949	85.0%
W Total	949			**85.0%**
PDB Name	750	27	976	87.5%
PDB Website	564	20	996	89.2%
PDB A/K/A	341	12	1,008	90.3%
PDB IRR	279	2	1,010	90.5%
PDB Total	843			**90.5%**
Manual	106	106	1,116	**100%**
Total		**1,116**		

85.0% validated using WHOIS.

+5.5% gain using PeeringDB.

+9.5% gain using Manual inspection.

1,116 of 1,147 (97.3%) are a valid name for the ASN

Limitation: unparseable convention

Hostname	ASN	
akamai.plix.pl	20940	(A)
cloudflare.plix.pl	13335	(B)
m247.plix.pl	9009	(C)
nask.plix.pl	204679	(D)
nask2.plix.pl	204679	(E)
oath2.plix.pl	10310	(F)
p4.plix.pl	39603	(G)

Name Dictionary
akamai:20940
cloudflare:13335
m247:9009
nask:204679
oath:10310
p4:39603

Not all operators use a convention that is suited to being parsed with a regex.

Limitation: unparseable convention

Hostname	ASN	
akamai.plix.pl	20940	(A)
cloudflare.plix.pl	13335	(B)
m247.plix.pl	9009	(C)
nask.plix.pl	204679	(D)
nask2.plix.pl	204679	(E)
oath2.plix.pl	10310	(F)
p4.plix.pl	39603	(G)

Name Dictionary

akamai:20940
cloudflare:13335
m247:9009
nask:204679
oath:10310
p4:39603

		TP	FN
RE1	^([a-z\d]+)\.plix\.pl$	A,B,C,D,G	E,F
RE2	^([a-z]+)\d+\.plix\.pl$	E,F	A,B,C,D,G

Not all operators use a convention that is suited to being parsed with a regex.

Hostnames A-D and G indicate primary connections by those ASes to PLIX.

Note: “m247” and “p4”

Limitation: unparseable convention

Hostname	ASN		Name Dictionary
akamai.plix.pl	20940	(A)	akamai:20940
cloudflare.plix.pl	13335	(B)	cloudflare:13335
m247.plix.pl	9009	(C)	m247:9009
nask.plix.pl	204679	(D)	nask:204679
nask2.plix.pl	204679	(E)	oath:10310
oath2.plix.pl	10310	(F)	p4:39603
p4.plix.pl	39603	(G)	

		TP	FN
RE1	^([a-z\d]+)\.plix\.pl$	A,B,C,D,G	E,F
RE2	^([a-z]+)\d+\.plix\.pl$	E,F	A,B,C,D,G

Not all operators use a convention that is suited to being parsed with a regex.

Hostnames E and F indicate secondary connections by "nask" and "oath" to PLIX.

Limitation: unparseable convention

Hostname	ASN	
akamai.plix.pl	20940	(A)
cloudflare.plix.pl	13335	(B)
m247.plix.pl	9009	(C)
nask.plix.pl	204679	(D)
nask2.plix.pl	204679	(E)
oath2.plix.pl	10310	(F)
p4.plix.pl	39603	(G)

Name Dictionary
akamai:20940
cloudflare:13335
m247:9009
nask:204679
oath:10310
p4:39603

		TP	FN
RE1	^([a-z\d]+)\.plix\.pl$	A,B,C,D,G	E,F
RE2	^([a-z]+)\d+\.plix\.pl$	E,F	A,B,C,D,G

Not all operators use a convention that is suited to being parsed with a regex.

Impossible to distinguish "2" for secondary connections, from digits that appear at end of "m247" and "p4" network names

Limitation: Network Name re-use

Hostname	ASN	Country
vodafone-level3-sanjose.level3.net	1273	UK
vodafone.sjc03.atlas.cogentco.com	1273	UK
vodafone.interxiondus1.nl-ix.net	3209	NL
vodafone1.ape.nzix.net	9500	NZ
vodafone.ronix.ro	12302	RO
vodafone.mix-it.net	30722	IT

Our method infers a single mapping between a network name and an ASN.

Independently operated networks with their own ASN may use the same network name in different countries.

Summary

- **We design and implement a method that automatically**
  - **learns regexes** that extract ***network names*** from hostnames,
  - **learns dictionary** that maps ***network names*** to their ***ASN***
- **We publicly release**
  - **the source code** implementation as part of ***Hoiho***,
    (Hoiho: Holistic Orthography of Internet Hostname Observations)
  - **the inferred naming conventions**
    - https://www.caida.org/tools/measurement/scamper/
    - https://www.caida.org/publications/papers/2021/hoiho-asnames/

Hoiho: Yellow-eyed penguin

Image: Brent Beaven
Department of Conservation (New Zealand)

Acknowledgments

- We thank Young Hyun and Ryan Koga for assistance with the ITDK, and the anonymous reviewers for their feedback.

- This work was partly supported by U.S. NSF awards CNS-2105393, 1925729, 1901517, and OAC-1724853, and the U.S. DoD Defense Advanced Research Projects Agency under CA-HR001120200014.

- It does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

Learning Regexes to Extract Network Names from Hostnames

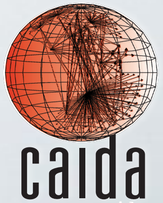
Matthew Luckie

University of Waikato

Alexander Marder
Bradley Huffaker
k claffy

CAIDA / UC San Diego

AINTEC 2021



BACKUP SLIDES

Why not use WHOIS or PeeringDB?

- Primary goal: have algorithm be self-contained
- Secondary issue: operators use names for networks, the network might **rebrand**, but hostname contains old name.

W OrgName: Orange S.A.
W ASName: Opentransit
PDB Name: Orange
PDB Website: https://wholesalesolutions.orange.com/
PDB A/K/A: Opentransit - IP Transit 5511
PDB IRR: AS-OPENTRANSIT

E.g., WHOIS (W) and PeeringDB (PDB) refer to "Opentransit" or "Orange" but hostnames contain historical name **"France Telecom" (FT)**

Why not use WHOIS or PeeringDB?

- Primary goal: have algorithm be self-contained

- Secondary issue: operators can abbreviate arbitrarily.
  How would we use external dictionary?

zayo.**vdms**.mpr4.atl6.us.zip.zayo.com (I)
verizondigitalmedia-com.customer.alter.net (J)
edgecast-gw.customer.alter.net (K)
verizondms.jfk10.atlas.cogentco.com (L)

The corresponding routers are operated by AS 15133: Edgecast.
Also known as "Verizon Digital Media Services"

W OrgName: MCI Communications Services, Inc. d/b/a Verizon Business
W ASName: EDGECAST
PDB Name: Verizon Digital Media Services (EdgeCast Networks)
PDB Website: https://www.verizondigitalmedia.com/
PDB A/K/A: EdgeCast Networks
PDB IRR: AS-EDGECAST

Phase 4: Refine Regexes

Hostname	ASN	
akamai.bix.bg	20940	(A)
cloudflare.bix.bg	13335	(B)
evolink.bix.bg	8262	(C)
evolink-b.bix.bg	8262	(D)
mitkocom.bix.bg	35761	(E)
mitkocom-b.bix.bg	35761	(F)

Name Dictionary

akamai:20940
cloudflare:13335
evolink:8262
mitkocom:35761

RE1

^([^\.]+)\.bix\.bg$

TP: A, B, C, E

RE2

^([^-]+)-[^\.]+\.bix\.bg$

TP: D, F

First set: network name is the only string in the hostname to the left of the suffix.

Phase 4: Refine Regexes

Hostname	ASN	
akamai.bix.bg	20940	(A)
cloudflare.bix.bg	13335	(B)
evolink.bix.bg	8262	(C)
evolink-b.bix.bg	8262	(D)
mitkocom.bix.bg	35761	(E)
mitkocom-b.bix.bg	35761	(F)

Name Dictionary

akamai:20940
cloudflare:13335
evolink:8262
mitkocom:35761

RE1
^([^\.]+)\.bix\.bg$
TP: A, B, C, E

RE3
^([a-z]+)\.bix\.bg$
TP: A, B, C, E

RE2
^([^-]+)-[^\.]+\.bix\.bg$
TP: D, F

First set: network name is the only string in the hostname to the left of the suffix.

Phase 4: Refine Regexes

Hostname	ASN	
akamai.bix.bg	20940	(A)
cloudflare.bix.bg	13335	(B)
evolink.bix.bg	8262	(C)
evolink-b.bix.bg	8262	(D)
mitkocom.bix.bg	35761	(E)
mitkocom-b.bix.bg	35761	(F)

Name Dictionary

akamai:20940
cloudflare:13335
evolink:8262
mitkocom:35761

RE1
^([^\.]+)\.bix\.bg$
TP: A, B, C, E

RE3
^([a-z]+)\.bix\.bg$
TP: A, B, C, E

RE2
^([^-]+)-[^\.]+\.bix\.bg$
TP: D, F

RE4
^([a-z]+)-b\.bix\.bg$
TP: D, F

Second set: network name appears prior to "-b" — indicating a secondary connection

Phase 4: Refine Regexes

Hostname	ASN	
akamai.bix.bg	20940	(A)
cloudflare.bix.bg	13335	(B)
evolink.bix.bg	8262	(C)
evolink-b.bix.bg	8262	(D)
mitkocom.bix.bg	35761	(E)
mitkocom-b.bix.bg	35761	(F)

Name Dictionary

akamai:20940
cloudflare:13335
evolink:8262
mitkocom:35761

RE1
^([^\.]+)\.bix\.bg$
TP: A, B, C, E

RE3
^([a-z]+)\.bix\.bg$
TP: A, B, C, E

RE2
^([^-]+)-[^\.]+\.bix\.bg$
TP: D, F

RE4
^([a-z]+)-b\.bix\.bg$
TP: D, F

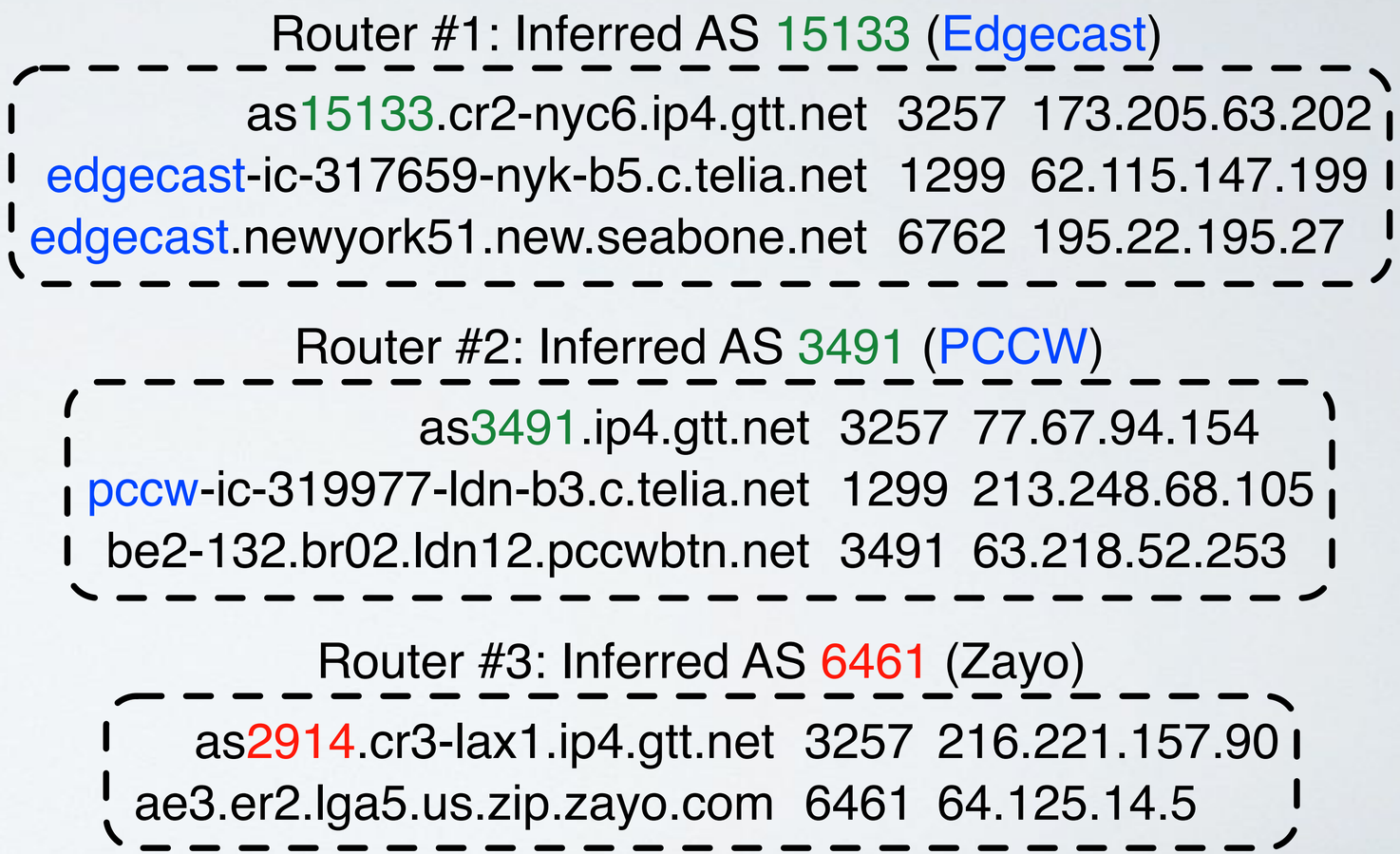
RE3 RE4
^([a-z]+)\.bix\.bg$
^([a-z]+)-b\.bix\.bg$

RE5
^([a-z]+)(?:-b)?\.bix\.bg$
TP: A, B, C, D, E, F

Emit a regex that captures that the network name can be followed by an optional "-b"

Our approach: use information in hostnames

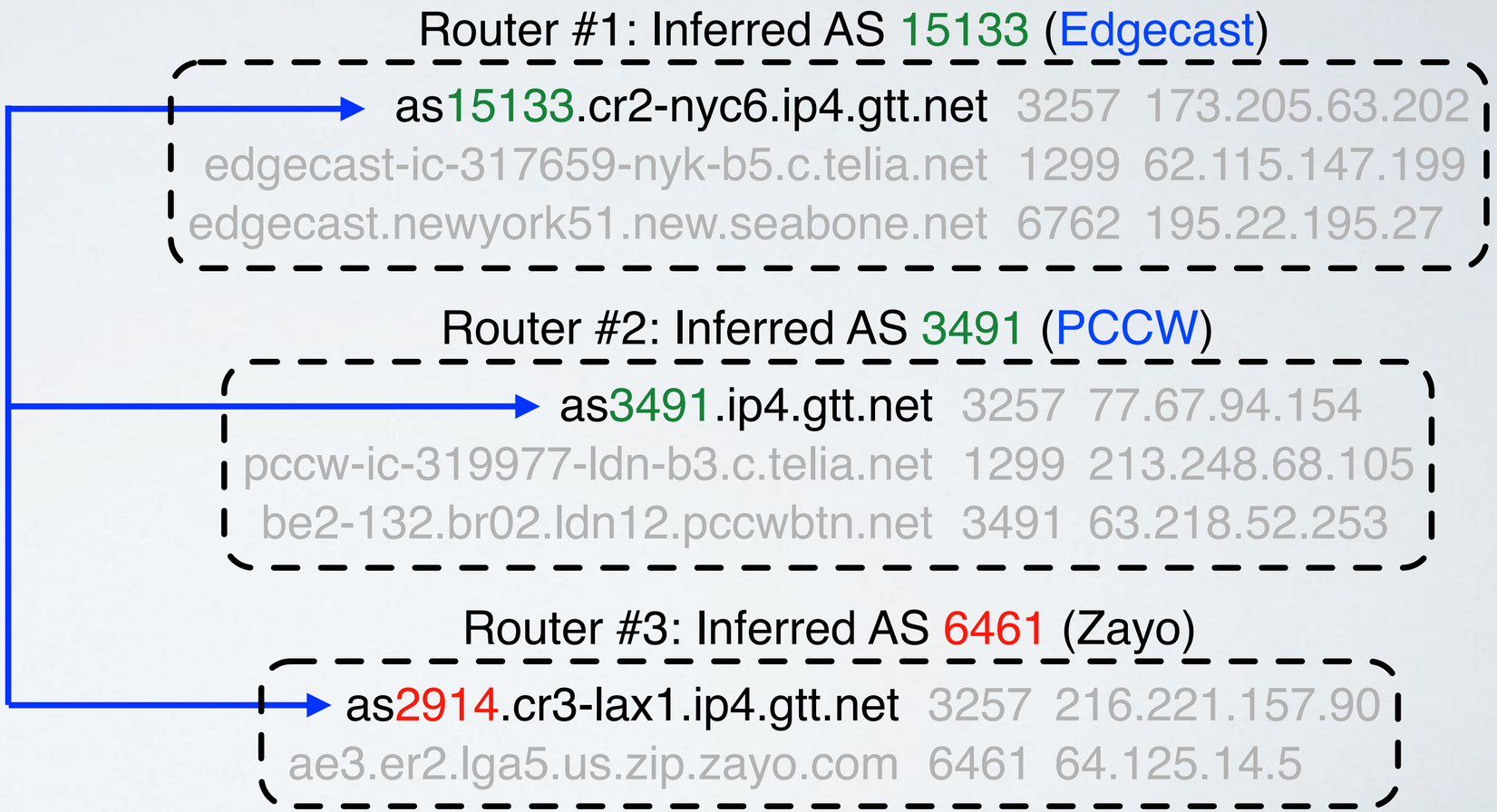
Some operators embed information in hostnames because it helps them debug their networks



Our approach: use information in hostnames

Some operators embed the **ASN** of the neighbor network using the address.

e.g. gtt.net



Luckie et al., IMC 2020: Learning to Extract and Use ASNs in Hostnames.

Our approach: use information in hostnames

Some operators embed the **name** of the neighbor network using the address.

e.g. telia.net, seabone.net