Learning to Extract Geographic Information from Internet Router Hostnames

● Matthew Luckie
□ Bradley Huffaker
□ Alexander Marder
□ Zachary Bischof
● Marianne Fletcher
□ k claffy

CoNEXT 2021

University of Waikato ●
□ CAIDA / UC San Diego

Motivation: Where are these routers located?

Router #1	154.54.9.6 173.205.55.118 206.111.0.201
Router #2	154.54.12.54 129.250.193.162 64.125.14.239
Router #3	109.200.218.13 4.14.228.118 168.143.105.162
Router #4	216.66.14.186 4.69.219.110 195.22.206.99

Motivation: Where are these routers located?

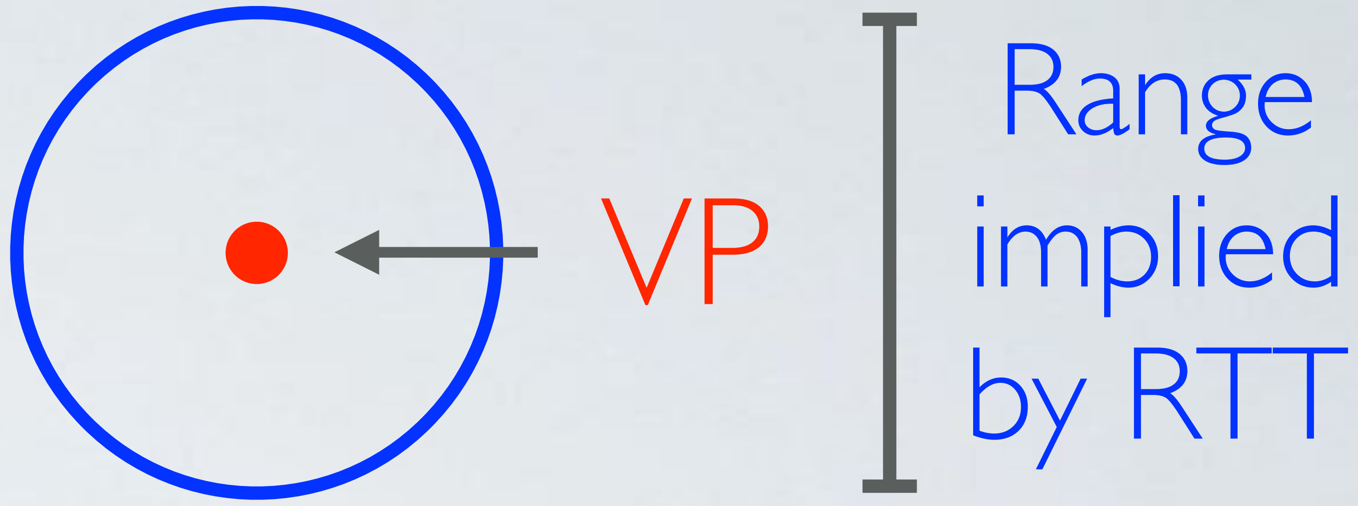
Router #1	154.54.9.6 173.205.55.118 206.111.0.201
Router #2	154.54.12.54 129.250.193.162 64.125.14.239
Router #3	109.200.218.13 4.14.228.118 168.143.105.162
Router #4	216.66.14.186 4.69.219.110 195.22.206.99

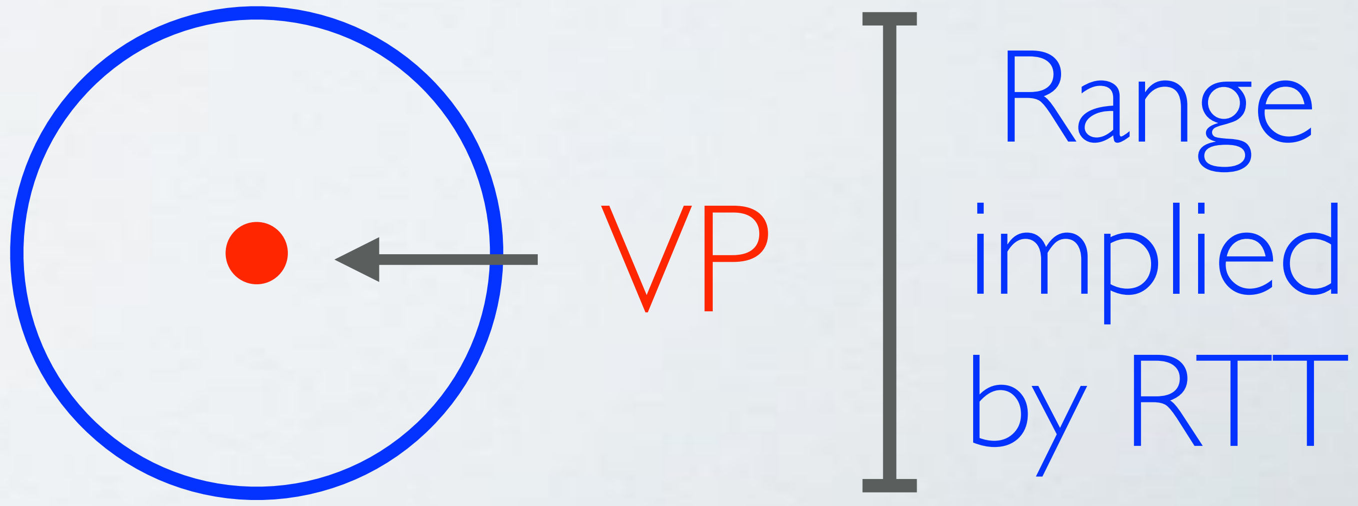
One approach: delay measurements.

System is located within distance implied by observed RTT from a known location.

Motivation: Where are these routers located?

Router #1	4ms from iad (Dulles, VA, US)	154.54.9.6 173.205.55.118 206.111.0.201	~400km
Router #2	3ms from iad (Dulles, VA, US)	154.54.12.54 129.250.193.162 64.125.14.239	~300km
Router #3	3ms from cgs (College Park, MD, US)	109.200.218.13 4.14.228.118 168.143.105.162	~300km
Router #4	4ms from cgs (College Park, MD, US)	216.66.14.186 4.69.219.110 195.22.206.99	~400km





Motivation: Where are these routers located?

Router #1	4ms from iad (Dulles, VA, US)	154.54.9.6 173.205.55.118 206.111.0.201	~400km
Router #2	3ms from iad (Dulles, VA, US)	154.54.12.54 129.250.193.162 64.125.14.239	~300km
Router #3	3ms from cgs (College Park, MD, US)	109.200.218.13 4.14.228.118 168.143.105.162	~300km
Router #4	4ms from cgs (College Park, MD, US)	216.66.14.186 4.69.219.110 195.22.206.99	~400km

No further than 300-400km from systems in the Washington D.C. area.

New York? New Jersey? Philadelphia? D.C. area?

Substantial work requires accurate router geolocation

Internet resilience to natural disasters, optimality of paths, etc.

- *iPlane*: An Information Plane for Distributed Services
- On inferring regional AS topologies
- Uncovering Performance Differences among Backbone ISPs with Netdiff
- Measuring and Evaluating Large-Scale CDNs
- Nation-State Routing: Censorship, Wiretapping, and BGP
- Packet Caches on Routers: The Implications of Universal Redundant Traffic Elimination
- Detecting Traffic Differentiation in Backbone ISPs with NetPolice
- Geography and Routing in the Internet
- A DISTRIBUTED SYSTEM FOR LARGE-SCALE GEOLOCALIZATION OF INTERNET HOSTS
- Out of Sight, Not Out of Mind - A User-View on the Criticality of the Submarine Cable Network
- Effective Diagnosis of Routing Disruptions from End Systems
- Geographic Locality of IP Prefixes
- Residential Links Under the Weather
- Solar Superstorms: Planning for an Internet Apocalypse

Selection of work including SIGCOMM papers published in 2019 and 2021

Intuition: Naming Conventions

Router #1	xo.iad02.atlas.cogentco.com as2828.was14.ip4.gtt.net te9-2-0d0.cir1.ashburn-va.us.xo.net
Router #2	vodafone.iad02.atlas.cogentco.com ae-0.vodafone.asbnva02.us.bb.gin.ntt.net zayo.vodafone.er2.iad10.us.zip.zayo.com
Router #3	usqas1-rt002i.i3d.net interactive.edge1.washington111.level3.net ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net
Router #4	level3-as3356.e0-51.switch2.ash1.he.net ae-1-3510.edge1.washington111.level3.net level3.ashburn2.ash.seabone.net

Hostnames suggest Washington D.C. area.

Goal: build a system that learns conventions that each operator uses to encode geohints.

Challenge: operators use different conventions

Router #1

xo.iad02.atlas.cogentco.com
as2828.was14.ip4.gtt.net
te9-2-0d0.cir1.ashburn-va.us.xo.net

Router #2

vodafone.iad02.atlas.cogentco.com
ae-0.vodafone.asbnva02.us.bb.gin.ntt.net
zayo.vodafone.er2.iad10.us.zip.zayo.com

Router #3

usqas1-rt002i.i3d.net
interactive.edge1.washington111.level3.net
ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net

Router #4

level3-as3356.e0-51.switch2.ash1.he.net
ae-1-3510.edge1.washington111.level3.net
level3.ashburn2.ash.seabone.net

Operators can choose their own convention.

We need to accommodate them all.

Our method inferred conventions for
1023 suffixes w/ IPv4 routers
241 suffixes w/ IPv6 routers

Challenge: operators use different dictionaries

Router #1

- xo.iad02.atlas.cogentco.com
- as2828.was14.ip4.gtt.net
- te9-2-0d0.cir1.ashburn-va.us.xo.net

Router #2

- vodafone.iad02.atlas.cogentco.com
- ae-0.vodafone.asbnva02.us.bb.gin.ntt.net
- zayo.vodafone.er2.iad10.us.zip.zayo.com

Router #3

- usqas1-rt002i.i3d.net
- interactive.edge1.washington111.level3.net
- ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net

Router #4

- level3-as3356.e0-51.switch2.ash1.he.net
- ae-1-3510.edge1.washington111.level3.net
- level3.ashburn2.ash.seabone.net

Most common: operators embed IATA airport code of closest airport.

Airport codes are unique.

Challenge: operators use different dictionaries

Router #1
- xo.iad02.atlas.cogentco.com
- as2828.was14.ip4.gtt.net
- te9-2-0d0.cir1.ashburn-va.us.xo.net

Router #2
- vodafone.iad02.atlas.cogentco.com
- ae-0.vodafone.asbnva02.us.bb.gin.ntt.net
- zayo.vodafone.er2.iad10.us.zip.zayo.com

Router #3
- usqas1-rt002i.i3d.net
- interactive.edge1.washington111.level3.net
- ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net

Router #4
- level3-as3356.e0-51.switch2.ash1.he.net
- ae-1-3510.edge1.washington111.level3.net
- level3.ashburn2.ash.seabone.net

Also common: operators embed place names.

Challenge: at least 27 populated places named "Washington" 4 named "Ashburn"

Challenge: operators use different dictionaries

Router #1	xo.iad02.atlas.cogentco.com as2828.was14.ip4.gtt.net te9-2-0d0.cir1.ashburn-va.us.xo.net
Router #2	vodafone.iad02.atlas.cogentco.com ae-0.vodafone.asbnva02.us.bb.gin.ntt.net zayo.vodafone.er2.iad10.us.zip.zayo.com
Router #3	usqas1-rt002i.i3d.net interactive.edge1.washington111.level3.net ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net
Router #4	level3-as3356.e0-51.switch2.ash1.he.net ae-1-3510.edge1.washington111.level3.net level3.ashburn2.ash.seabone.net

Also common: operators embed (portions of) a CLLI code.

Chars	Meaning	e.g.
4	Place	asbn
2	State/country	va
5	various	

asbnva = Ashburn, VA, US

Challenge: operators use different dictionaries

Router #1
- xo.iad02.atlas.cogentco.com
- as2828.was14.ip4.gtt.net
- te9-2-0d0.cir1.ashburn-va.us.xo.net

Router #2
- vodafone.iad02.atlas.cogentco.com
- ae-0.vodafone.asbnva02.us.bb.gin.ntt.net
- zayo.vodafone.er2.iad10.us.zip.zayo.com

Router #3
- usqas1-rt002i.i3d.net
- interactive.edge1.washington111.level3.net
- ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net

Router #4
- level3-as3356.e0-51.switch2.ash1.he.net
- ae-1-3510.edge1.washington111.level3.net
- level3.ashburn2.ash.seabone.net

UN LOCODEs are less common, and not always human readable.

Chars	Meaning	e.g.
2	Country	us
3	Place	qas

qas = Ashburn, VA, US

Challenge: operators use different dictionaries

Router #1

xo.iad02.atlas.cogentco.com
as2828.was14.ip4.gtt.net
te9-2-0d0.cir1.ashburn-va.us.xo.net

Router #2

vodafone.iad02.atlas.cogentco.com
ae-0.vodafone.asbnva02.us.bb.gin.ntt.net
zayo.vodafone.er2.iad10.us.zip.zayo.com

Router #3

usqas1-rt002i.i3d.net
interactive.edge1.washington111.level3.net
ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net

Router #4

level3-as3356.e0-51.switch2.ash1.he.net
ae-1-3510.edge1.washington111.level3.net
level3.ashburn2.ash.seabone.net

Some operators helpfully embed country or state codes in their hostnames.

i.e., we know that "ashburn" refers to the one in Virginia, US.

Challenge: operators deviate from dictionaries

Router #1
xo.iad02.atlas.cogentco.com
as2828.was14.ip4.gtt.net
te9-2-0d0.cir1.ashburn-va.us.xo.net

Router #2
vodafone.iad02.atlas.cogentco.com
ae-0.vodafone.asbnva02.us.bb.gin.ntt.net
zayo.vodafone.er2.iad10.us.zip.zayo.com

Router #3
usqas1-rt002i.i3d.net
interactive.edge1.washington111.level3.net
ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net

Router #4
level3-as3356.e0-51.switch2.ash1.he.net
ae-1-3510.edge1.washington111.level3.net
level3.ashburn2.ash.seabone.net

he.net and seabone.net both use "ash" to mean "Ashburn, VA, US".

"ash" is the IATA code for "Nashua, NH, US"

Implication: must learn per-suffix dictionaries

Challenge: abbreviations are lossy

Contributions of this work

- **We design and implement a method that automatically**
  - **learns regexes** that extract geohints from hostnames,
  - **learns new geohints** when operators deviate from the dictionary.
- **We publicly release**
  - **the source code** implementation as part of Hoiho,
    (Hoiho: Holistic Orthography of Internet Hostname Observations)
  - **the inferred naming conventions** and a utility to apply them.

Hoiho: Yellow-eyed penguin

- https://www.caida.org/tools/measurement/scamper/
- https://www.caida.org/publications/papers/2021/hoiho/

Image: Brent Beaven
Department of Conservation (New Zealand)

Key Results

- For an **August 2020** set of **2.56M** routers with IPv4 addresses
  - **8.8%** had hostnames containing apparent geohints
  - Our method inferred naming conventions for **906 suffixes** that extracted geohints from **86.8%** of these routers

    147 (**38.2%**) of 461 suffixes deviated from IATA dictionary: deviation from dictionary was common → IATA

  - We evaluated our method on four sets of routers, with **IPv4** and **IPv6** routers, to infer **1023** and **241** conventions, respectively, for these routers.

Type	**Freq**
IATA	461
City	372
CLLI prefix	96
LOCODE	10
Facility	2

Selected Related Work

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- HLOC: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

Selected Related Work: **undns**

- **undns**: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- HLOC: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

Hand-crafted regexes built by manually interpreting hostnames. Hand-crafted rules to interpret extracted output.

```
^be-\d+\.(cor|bdr)\d+\.([a-z]{3})\d+\.[a-z]{2,3}\.vocus\.net\.au$
```

```
be-102.cor01.per02.wa.vocus.net.au
be-103.cor01.per02.wa.vocus.net.au

be-102.cor02.mel07.vic.vocus.net.au
be-151.cor02.mel07.vic.vocus.net.au

be-100.bdr01.syd03.nsw.vocus.net.au
be-101.bdr01.syd03.nsw.vocus.net.au
```

Example routers

```
type=1 {
  cor "backbone"
  bdr "gateway"
}
loc=2 {
  per "Perth, Australia"
  mel "Melbourne, Australia"
  syd "Sydney, Australia"
}
```

Selected Related Work: **CBG**

- undns: (SIGCOMM 2002)
- **CBG**: (IMC 2004)
- DRoP: (CCR 2014)
- HLOC: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

(Figure 2 of "Constraint Based Geolocation of Internet Hosts")

CBG infers a system is located at the **centroid** of distance constraints built using delay measurements from vantage points with known locations.

Selected Related Work: **DRoP**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- **DRoP**: (CCR 2014)
- HLOC: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

Automatically built regexes that extract apparent geohints from router hostnames, using RTT constraints collected by traceroute

Matched

Unmatched

den1-core-01-ae1.360.net
den1-core-01-xe-1-1-0.360.net

den1-core-02-ae1.360.net
den1-core-02-xe-0-1-0.360.net

lax1-edge-01-1-1-1.360.net

sea1-edge-02-lag1.360.net

pdx2-access-01-1-1-2.360.net

^([a-z]{3})([^a-z]+[a-z]+[0-9]*){2}\.360\.net$

Limitation: RTT constraints collected by traceroute do not provide tight constraints. Multiple works report that more DRoP-inferred locations are wrong than correct.

Selected Related Work: **HLOC**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- **HLOC**: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

For each hostname, identify possible geohints and collect RTT measurements from *nearby* VPs to confirm.

Limitations: confirmation bias. Does not attempt to learn geocodes not in input dictionaries.

Selected Related Work: **HLOC**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- **HLOC**: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

de-cix1.rt.act.fkt.de.retn.net

For each hostname, identify possible geohints and collect RTT measurements from *nearby* VPs to confirm.

Limitations: confirmation bias. Does not attempt to learn geocodes not in input dictionaries.

Selected Related Work: **HLOC**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- **HLOC**: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

de-cix1.rt.act.fkt.de.retn.net

cix
IATA code
(Chiclayo, PE)

act
IATA code
(Waco, TX, US)

For each hostname, identify possible geohints and collect RTT measurements from *nearby* VPs to confirm.

Limitations: confirmation bias.
Does not attempt to learn geocodes not in input dictionaries.

Selected Related Work: **HLOC**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- **HLOC**: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

de-cix1.rt.act.fkt.de.retn.net

HLOC does not consider fkt.de (Frankfurt, HE, DE) because fkt is not in the dictionary

For each hostname, identify possible geohints and collect RTT measurements from *nearby* VPs to confirm.

Limitations: confirmation bias. Does not attempt to learn geocodes not in input dictionaries.

Selected Related Work: **Hoiho**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- HLOC: (TMA 2017)
- **Hoiho**: (IMC 2019 + 2020)

be-102.cor01.per02.wa.vocus.net.au
be-103.cor01.per02.wa.vocus.net.au

be-102.cor02.mel07.vic.vocus.net.au
be-151.cor02.mel07.vic.vocus.net.au

be-100.bdr01.syd03.nsw.vocus.net.au
be-101.bdr01.syd03.nsw.vocus.net.au

^be-\d+\.([a-z]+\d+\.[a-z]+\d+\.[a-z]+)\.vocus\.net\.au$

Hoiho 2019: learn regexes that extract router names (strings shared across router interface hostnames unique to each router)

(Hoiho: Holistic Orthography of Internet Hostname Observations)

Selected Related Work: **Hoiho**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- DRoP: (CCR 2014)
- HLOC: (TMA 2017)
- **Hoiho**: (IMC 2019 + 2020)

as10083.cust.bdr02.syd04.nsw.vocus.net.au
as11086.bdr01.syd03.nsw.vocus.net.au
as45763.cust.bdr02.per02.wa.vocus.net.au
asn131476.cust.bdr01.syd01.nsw.vocus.net.au
asn131107.bdr01.bne03.qld.vocus.net.au
asn132712.bdr02.mel07.vic.vocus.net.au

^asn?(\d+)\.[^\.]+\..+\.vocus\.net\.au$

Hoiho 2020: learn regexes that extract router ownership information reported by operators in ASN tags

(Hoiho: Holistic Orthography of Internet Hostname Observations)

Overview of Our Geolocation Method in Hoiho

We licensed a database mapping *CLLI prefixes* to location names from iconectiv; we mapped these to lat/longs.

Overview of Our Geolocation Method in Hoiho

Focus in this talk on identifying apparent geohints, and interpreting new geohints.

See paper for rest of method detail.

Identify possible geohints with input dictionary

ae0-2.rt.irx.vie.at.retn.net
ae1-1.rt.irx.vie.at.retn.net

ae1-4.rt.evo.ams.nl.retn.net

ae0-4.rt.wbx.sea.us.retn.net

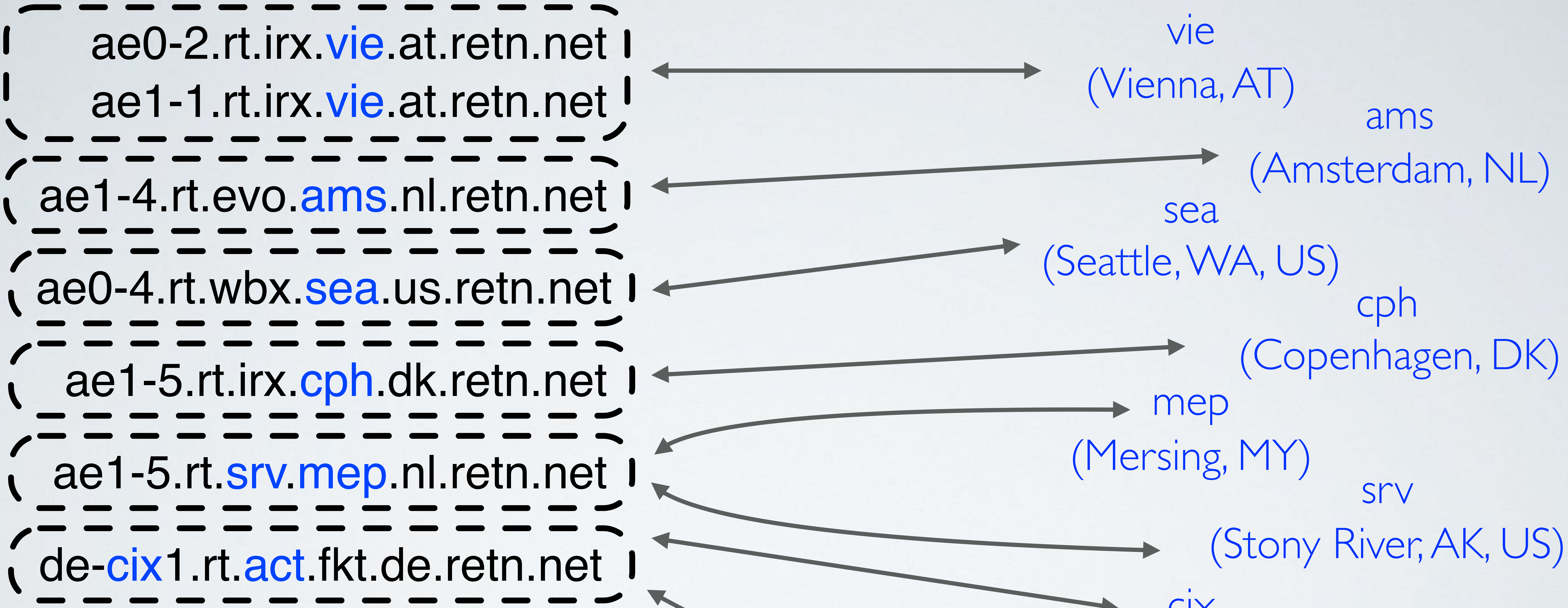
ae1-5.rt.irx.cph.dk.retn.net

ae1-5.rt.srv.mep.nl.retn.net

de-cix1.rt.act.fkt.de.retn.net

Consider sequences of alphabetical characters that are defined in the input dictionary.

Identify possible geohints with input dictionary



Consider sequences of alphabetical characters that are defined in the input dictionary.

Identify apparent geohints with RTT measurements

ae0-2.rt.irx.vie.at.retn.net
ae1-1.rt.irx.vie.at.retn.net

3ms feasible from VP in Vienna?

vie (Vienna, AT)

ae1-4.rt.evo.ams.nl.retn.net

2ms feasible from VP in Amsterdam?

ams (Amsterdam, NL)

ae0-4.rt.wbx.sea.us.retn.net

3ms feasible from VP in Seattle?

sea (Seattle, WA, US)

ae1-5.rt.irx.cph.dk.retn.net

13ms feasible from VP in Amsterdam?

cph (Copenhagen, DK)

ae1-5.rt.srv.mep.nl.retn.net

mep (Mersing, MY)

srv (Stony River, AK, US)

de-cix1.rt.act.fkt.de.retn.net

cix (Chiclayo, PE)

act (Waco, TX, US)

A string is an ***apparent geohint*** if measured RTTs are consistent with the location implied by the geohint

Identify apparent geohints with RTT measurements

- ae0-2.rt.irx.vie.at.retn.net
- ae1-1.rt.irx.vie.at.retn.net
  - 3ms feasible from VP in Vienna? → vie (Vienna, AT)
- ae1-4.rt.evo.ams.nl.retn.net
  - 2ms feasible from VP in Amsterdam? → ams (Amsterdam, NL)
- ae0-4.rt.wbx.sea.us.retn.net
  - 3ms feasible from VP in Seattle? → sea (Seattle, WA, US)
- ae1-5.rt.irx.cph.dk.retn.net
  - 13ms feasible from VP in Amsterdam? → cph (Copenhagen, DK)
- ae1-5.rt.srv.mep.nl.retn.net
  - **4ms** feasible from VP in Amsterdam? → **mep (Mersing, MY)**, **srv (Stony River, AK, US)**
- de-cix1.rt.act.fkt.de.retn.net
  - **8ms** feasible from VP in Zurich? → **cix (Chiclayo, PE)**, **act (Waco, TX, US)**

A string is an ***apparent geohint*** if measured RTTs are consistent with the location implied by the geohint

Identify apparent geohints with RTT measurements

- ae0-2.rt.irx.vie.at.retn.net / ae1-1.rt.irx.vie.at.retn.net ↔ vie (Vienna, AT)
- ae1-4.rt.evo.ams.nl.retn.net ↔ ams (Amsterdam, NL)
- ae0-4.rt.wbx.sea.us.retn.net ↔ sea (Seattle, WA, US)
- ae1-5.rt.irx.cph.dk.retn.net ↔ cph (Copenhagen, DK)
- ae1-5.rt.srv.mep.nl.retn.net
- de-cix1.rt.act.fkt.de.retn.net

A string is an ***apparent geohint*** if measured RTTs are consistent with the location implied by the geohint

Identify apparent geohints with RTT measurements

Hostname	Apparent geohint
ae0-2.rt.irx.vie.at.retn.net ae1-1.rt.irx.vie.at.retn.net	vie.**at** (Vienna, AT)
ae1-4.rt.evo.ams.nl.retn.net	ams.**nl** (Amsterdam, NL)
ae0-4.rt.wbx.sea.us.retn.net	sea.**us** (Seattle, WA, US)
ae1-5.rt.irx.cph.dk.retn.net	cph.**dk** (Copenhagen, DK)
ae1-5.rt.srv.mep.nl.retn.net	
de-cix1.rt.act.fkt.de.retn.net	

Note: these hostnames appear to include an ISO-3166 country-code.
Our method identifies this as something it should extract.

Build Regular Expressions to Extract Apparent Geohints

(see paper for details)

ae0-2.rt.irx.vie.at.retn.net
ae1-1.rt.irx.vie.at.retn.net

ae1-4.rt.evo.ams.nl.retn.net

ae0-4.rt.wbx.sea.us.retn.net

ae1-5.rt.irx.cph.dk.retn.net

ae1-5.rt.srv.mep.nl.retn.net

de-cix1.rt.act.fkt.de.retn.net

^.+\.([a-z]{3})\.([a-z]{2})\.retn\.net$

Build Regular Expressions to Extract Apparent Geohints

(see paper for details)

ae0-2.rt.irx.vie.at.retn.net
ae1-1.rt.irx.vie.at.retn.net

ae1-4.rt.evo.ams.nl.retn.net

ae0-4.rt.wbx.sea.us.retn.net

ae1-5.rt.irx.cph.dk.retn.net

ae1-5.rt.srv.mep.nl.retn.net

de-cix1.rt.act.fkt.de.retn.net

^.+\.([a-z]{3})\.([a-z]{2})\.retn\.net$

([a-z]{3}) → **IATA**

([a-z]{2}) → **CC**

Our method includes a *plan* for each regex: i.e., what each extraction represents.

Build Regular Expressions to Extract Apparent Geohints

(see paper for details)

Hostname		Geohint	Location
ae0-2.rt.irx.vie.at.retn.net	⟷	iata: **vie**, cc: **at**	
ae1-1.rt.irx.vie.at.retn.net	⟷	iata: **vie**, cc: **at**	**Vienna, AT**
ae1-4.rt.evo.ams.nl.retn.net	⟷	iata: **ams**, cc: **nl**	**Amsterdam, NL**
ae0-4.rt.wbx.sea.us.retn.net	⟷	iata: **sea**, cc: **us**	**Seattle, WA, US**
ae1-5.rt.irx.cph.dk.retn.net	⟷	iata: **cph**, cc: **dk**	**Copenhagen, DK**
ae1-5.rt.srv.mep.nl.retn.net			
de-cix1.rt.act.fkt.de.retn.net			

^.+\.([a-z]{3})\.([a-z]{2})\.retn\.net$

IATA **CC**

Build Regular Expressions to Extract Apparent Geohints

(see paper for details)

Hostname		Extracted	Location
ae0-2.rt.irx.vie.at.retn.net	⟷	iata: **vie**, cc: **at**	**Vienna, AT**
ae1-1.rt.irx.vie.at.retn.net	⟷	iata: **vie**, cc: **at**	
ae1-4.rt.evo.ams.nl.retn.net	⟷	iata: **ams**, cc: **nl**	**Amsterdam, NL**
ae0-4.rt.wbx.sea.us.retn.net	⟷	iata: **sea**, cc: **us**	**Seattle, WA, US**
ae1-5.rt.irx.cph.dk.retn.net	⟷	iata: **cph**, cc: **dk**	**Copenhagen, DK**
ae1-5.rt.srv.mep.nl.retn.net	⟷	iata: **mep**, cc: **nl**	**???**
de-cix1.rt.act.fkt.de.retn.net	⟷	iata: **fkt**, cc: **de**	**???**

```
^.+\.([a-z]{3})\.([a-z]{2})\.retn\.net$
```

IATA **CC**

Learn Geohints not in Dictionary

Consider abbreviations of RTT-consistent populated places

ae1-5.rt.srv.mep.nl.retn.net

4ms from Amsterdam, NL

Place	Population
★ **Me**p**p**el, DR, **NL**	30,697
Mep**p**en, DR, **NL**	305
Midd**e**lkoo**p**, UT, **NL**	370

de-cix1.rt.act.fkt.de.retn.net

8ms from Zurich, CH

Place	Population
★ **F**ran**k**fur**t** am Main, HE, **DE**	650,000
Fran**k**en**t**hal, RP, **DE**	47,438
Fal**k**ens**t**ein, **DE**	9,528
+ 5 other locations	

★ Place has a facility listed in PeeringDB

Name of candidate populated place must match first letter in abbreviation. Prefer places with known facilities, then places with higher population.

Validation of learned geohints against ground truth

aorta.net	as8218.eu	geant.net	gtt.net	he.net
3/4 (75%)	3/3 (100%)	8/8 (100%)	12/12 (100%)	4/4 (100%)

ntt.net	retn.net	seabone.net	tfbnw.net	zayo.net
17/18 (94.4%)	25/34 (73.5%)	14/15 (93.3%)	2/14 (14.3%)	4/4 (100%)

Overall
92/117 (78.6%)

We obtained ground truth for the learned geohints from operators at 10 different networks.

Validation of learned geohints against ground truth

aorta.net	as8218.eu	geant.net	gtt.net	he.net
3/4 (75%)	3/3 (100%)	8/8 (100%)	12/12 (100%)	4/4 (100%)

ntt.net	retn.net	seabone.net	tfbnw.net	zayo.net
17/18 (94.4%)	25/34 (73.5%)	14/15 (93.3%)	2/14 (14.3%)	4/4 (100%)

Overall
92/117 (78.6%)

Overall, 78.6% of the learned geohints we validated identified the correct place.

Validation of learned geohints against ground truth

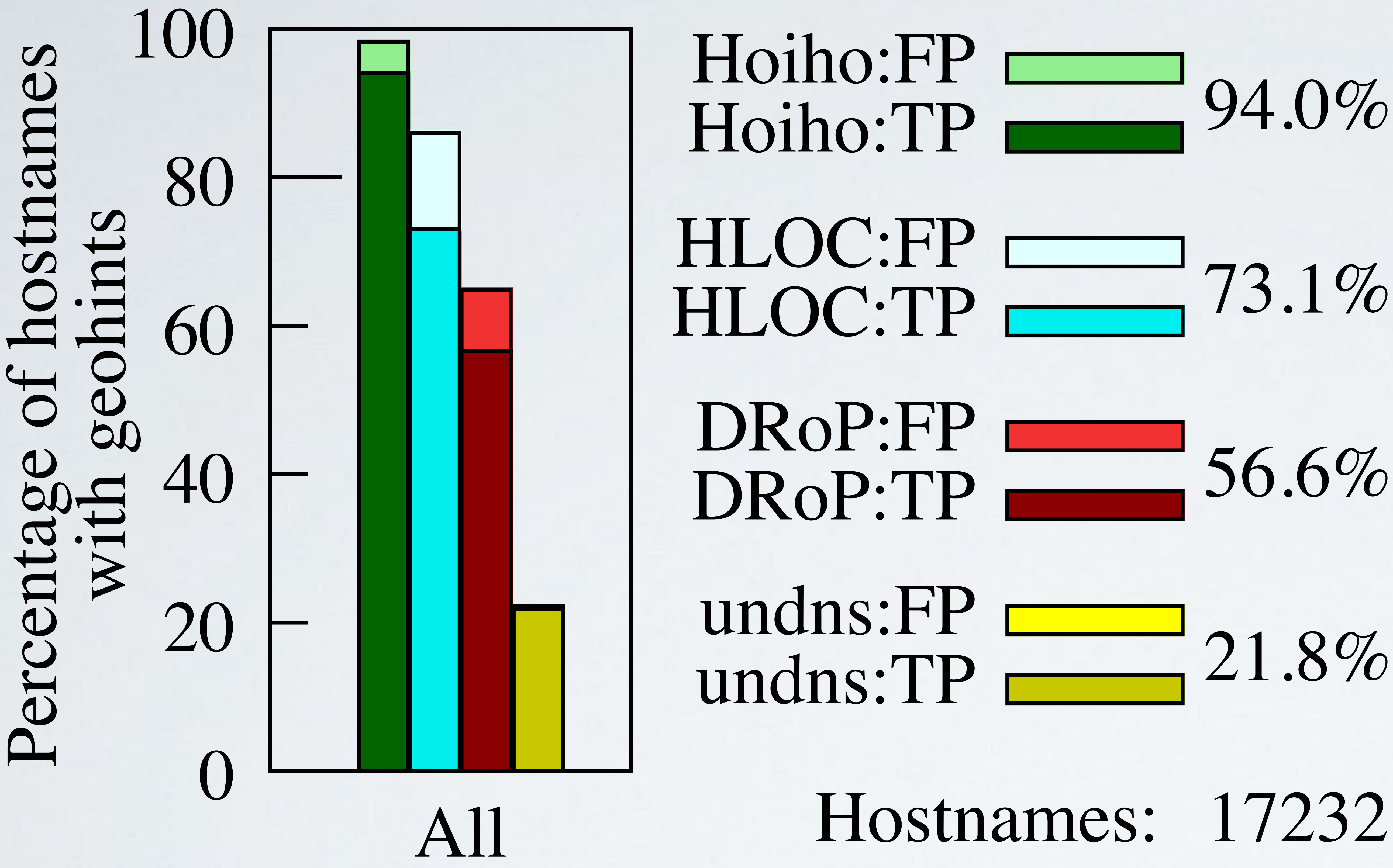
aorta.net	as8218.eu	geant.net	gtt.net	he.net
3/4 (75%)	3/3 (100%)	8/8 (100%)	12/12 (100%)	4/4 (100%)

ntt.net	retn.net	seabone.net	tfbnw.net	zayo.net
17/18 (94.4%)	25/34 (73.5%)	14/15 (93.3%)	2/14 (14.3%)	4/4 (100%)

Overall
92/117 (78.6%)

Outlier: Facebook, which places datacenter facilities in low-population locations that are not used as peering facilities (not in PeeringDB)

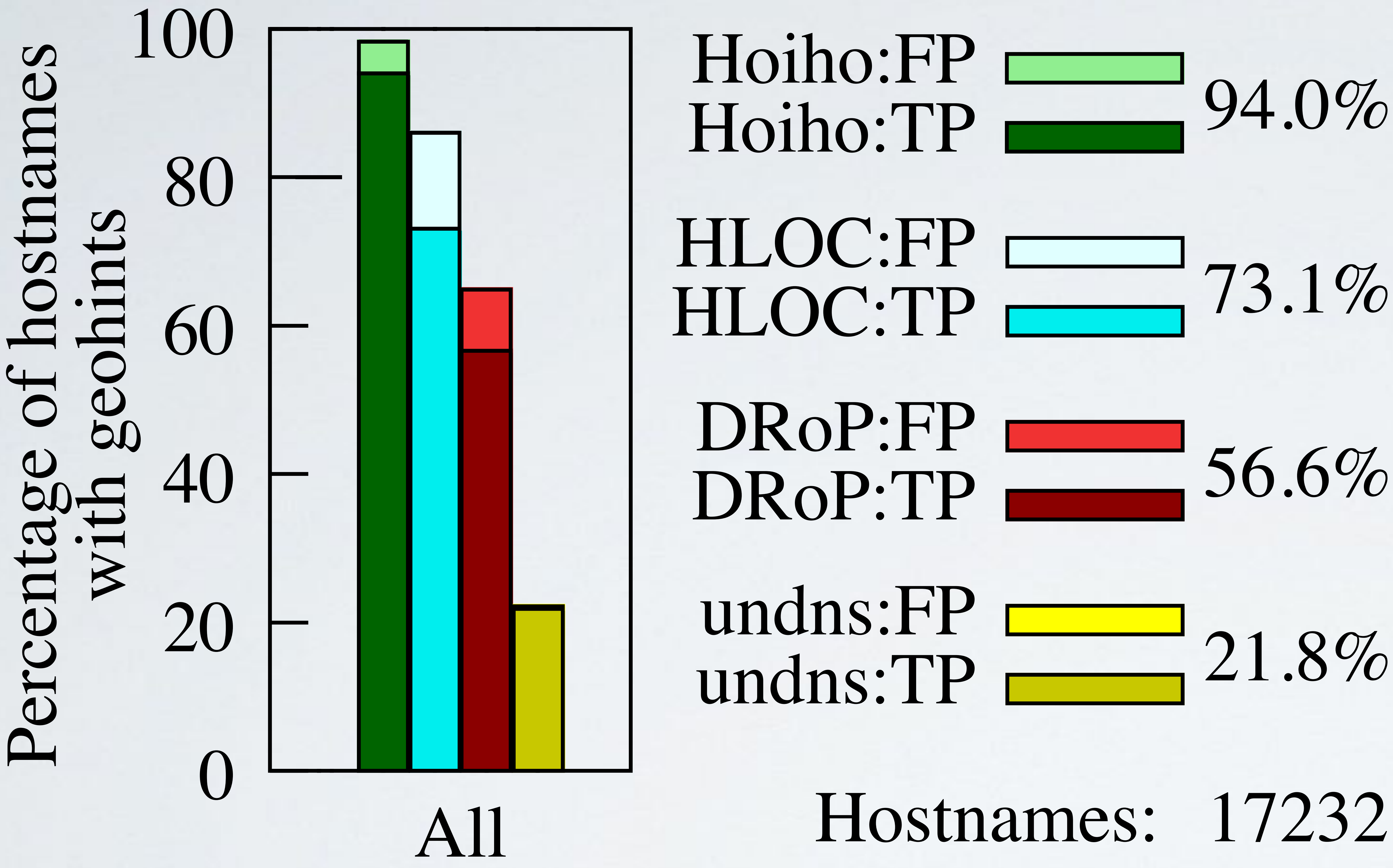
Validation of conventions against ground truth



- **Our method in Hoiho** inferred the correct location for 94.0% of hostnames across 14 suffixes.
- **DRoP and undns** coverage is lower as their conventions are old
- **HLOC and DRoP** FPs are because they don't learn custom geohints
- **undns** also missing location mappings

Note: gap between top of bar and 100% are false negatives: geohints missed by a method.

Validation of conventions against ground truth



Method	PPV
Hoiho	95.6%
HLOC	85.1%
DRoP	87.2%
undns	98.3%

Though undns has lowest coverage, it has highest PPV:TP/(TP+FP). The locations it has in its dictionary are generally correct.

Limitation: not all operators use an easily parsed convention

VP	RTT	Hostname
atl, us	7ms	atnga00002cce9-irb-2.infra.cdn.att.net
ord, us	9ms	bcvoh00002cce9-irb-2.infra.cdn.att.net
dal, us	5ms	dlltx00001cce9-irb-2.infra.cdn.att.net
jfk, us	1ms	nycny00002cce9-irb-2.infra.cdn.att.net
dal, us	4ms	rd3tx00001cce9-ae120-100.infra.cdn.att.net
sjc, us	4ms	scaca00002cce9-ae120-200.infra.cdn.att.net

Dictionary:

- atnga: Atlanta, GA
- bcvoh: Brecksville, OH
- dlltx: Dallas, TX
- nycny: New York City, NY
- rd3tx: Richardson, TX
- scaca: Sacramento, CA

AT&T uses a convention with no punctuation between 3-letter abbreviation of place and 2-letter state code. 3-letter abbreviations are not based on airport codes and difficult even for a human to decipher.

Summary

- **We designed and implemented a method that automatically**
  - **learns regexes** that extract geohints from hostnames,
  - **learns new geohints** when operators deviate from the dictionary.
- **We publicly release**
  - **the source code** implementation as part of Hoiho,
    (Hoiho: Holistic Orthography of Internet Hostname Observations)
  - **the inferred naming conventions** and a utility to apply them.

- https://www.caida.org/tools/measurement/scamper/
- https://www.caida.org/publications/papers/2021/hoiho/

Method	Coverage
Hoiho	94.0%
HLOC	73.1%
DRoP	56.6%
undns	21.8%

Acknowledgements

- We thank Young Hyun for assistance with the ITDK, our shepherd Gareth Tyson, and the anonymous reviewers for their feedback.

- This work is 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-HR00112020014.

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

BACKUP SLIDES

High-level Approach

- Infer if an operator embeds information **identifying the location of the router** in PTR hostname records for router interfaces
- **Input**:
  - Mozilla public suffix list to identify where domains can be registered (.net, .org, .co.nz)
  - Hostnames for router interfaces observed by traceroute (PTR records)
  - Router alias inferences from MIDAR, mercator
  - RTT measurements using ICMP, UDP, and TCP pings
  - Geohint dictionary with IATA, ICAO, CLLI prefixes, LOCODEs, Towns, States, Countries
- **Output**: ***regular expressions*** that extract router geolocation, and a ***dictionary*** to interpret the ***geohints***.

Results: Coverage of Inferred Naming Conventions

Routers	August 2020 IPv4	March 2021 IPv4	November 2020 IPv6	March 2021 IPv6
total	2.56M	2.57M	559K	525K
with hostname	1.41M (55.0%)	1.39M (54.1%)	84K (15.1%)	84K (16.0%)
with apparent geohint	225K (8.8%)	220K (8.5%)	29K (5.3%)	31K (5.8%)
geolocated	195K (7.6%)	183K (7.1%)	26K (4.7%)	27K (5.2%)

We used CAIDA ITDKs where we simultaneously collected RTT samples from available CAIDA Archipelago Vantage Points. Our conventions extracted 83.4% - 89.6% of apparent geohints.

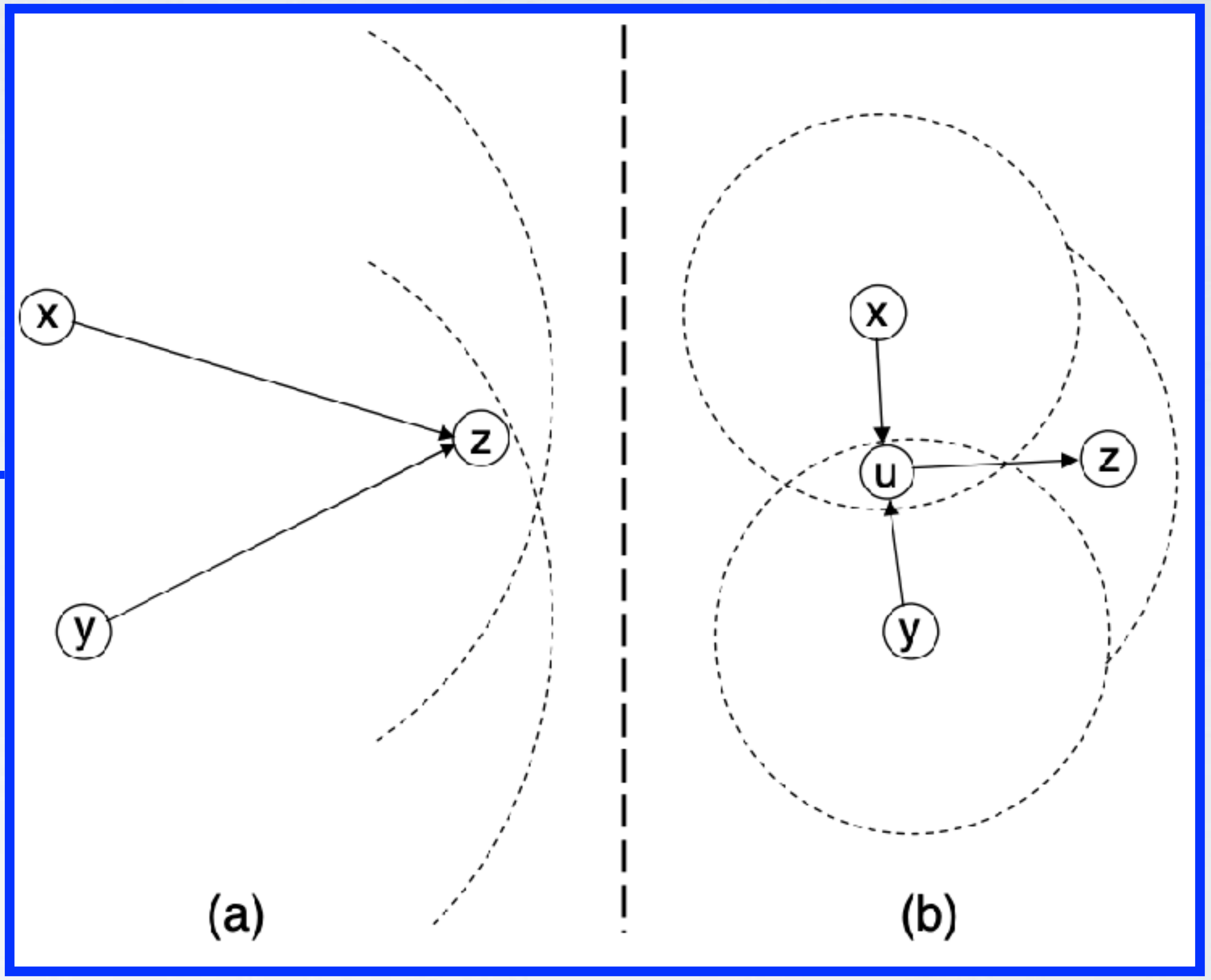
Validation of conventions with ground truth

	above.net	akamai.com	aorta.net	as8218.eu	geant.net	gtt.net	he.net	ntt.net	nysernet.net	retn.net	seabone.net	tfbnw.net	vrsn.net	zayo.com	All
Hoiho:	71.7%	99.3%	75.6%	99.6%	93.0%	95.5%	98.5%	95.5%	84.3%	78.7%	88.6%	90.3%	94.7%	96.7%	94.0%
HLOC:	63.3%	79.8%	47.3%	81.1%	63.7%	82.8%	77.2%	68.4%	3.9%	55.7%	78.9%	68.3%	5.3%	77.7%	73.1%
DRoP:	5.0%	0.0%	0.0%	0.0%	0.0%	0.0%	80.4%	89.5%	2.0%	48.6%	60.3%	82.8%	0.0%	79.6%	56.6%
Undns:	61.7%	0.0%	56.5%	0.0%	0.0%	0.0%	28.4%	72.1%	7.8%	0.0%	47.7%	0.0%	0.0%	0.0%	21.8%
Hostnames:	60	2403	131	472	270	1697	2121	3397	51	479	1238	4128	19	766	17232

Selected Related Work: **TBG**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- **TBG**: (IMC 2006)
- DRoP: (CCR 2014)
- HLOC: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)

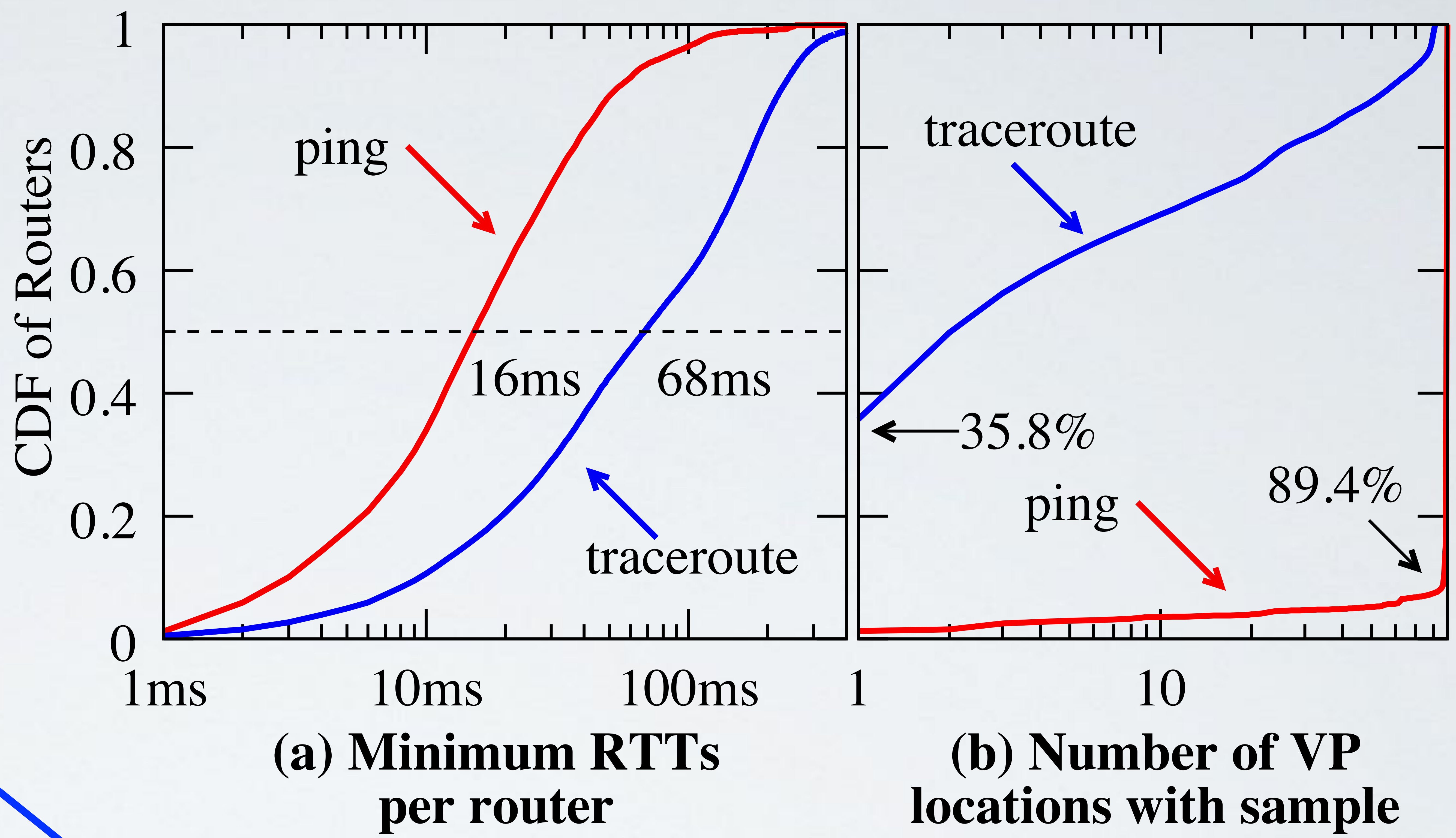
(Figure 6 of "Towards IP Geolocation Using Delay and Topology Measurements")



TBG adds topological constraints (U) to CBG — i.e., intermediate routers observed using traceroute that reduce the distance that Z could be from X/Y.

Selected Related Work: **DRoP**

- undns: (SIGCOMM 2002)
- CBG: (IMC 2004)
- **DRoP**: (CCR 2014)
- HLOC: (TMA 2017)
- Hoiho: (IMC 2019 + 2020)



Limitation: RTT constraints collected by traceroute do not provide tight constraints. Multiple works report that more DRoP-inferred locations are wrong than correct.

Key Results: Validation

- We compared our geolocation inferences and those made by other approaches with ground truth for hostnames in 14 suffixes.
  - Our method has the highest coverage (**94.0%**) and a PPV of **95.6%**
- We compared our learned geohints against ground truth from 10 suffixes with 117 suffix-specific geohints
  - **92/117 (78.6%)** correctly identified the corresponding location

Method	Coverage	PPV
Our Method	94.0%	95.6%
HLOC	73.1%	85.1%
DRoP	56.6%	87.2%
undns	21.8%	98.3%