Learning to Extract Geographic Information from Internet Router Hostnames

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

University of Waikato  CAIDA / UC San Diego

CoNEXT 2021
Motivation: Where are these routers located?

<table>
<thead>
<tr>
<th>Router #1</th>
<th>154.54.9.6</th>
<th>173.205.55.118</th>
<th>206.111.0.201</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router #2</td>
<td>154.54.12.54</td>
<td>129.250.193.162</td>
<td>64.125.14.239</td>
</tr>
</tbody>
</table>
Motivation: Where are these routers located?

One approach: delay measurements.

System is located within distance implied by observed RTT from a known location.
## Motivation: Where are these routers located?

<table>
<thead>
<tr>
<th>Router #1</th>
<th>4ms from iad (Dulles, VA, US)</th>
<th>154.54.9.6</th>
<th>~400km</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>173.205.55.118</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>206.111.0.201</td>
<td></td>
</tr>
<tr>
<td>Router #2</td>
<td>3ms from iad (Dulles, VA, US)</td>
<td>154.54.12.54</td>
<td>~300km</td>
</tr>
<tr>
<td></td>
<td></td>
<td>129.250.193.162</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>64.125.14.239</td>
<td></td>
</tr>
<tr>
<td>Router #3</td>
<td>3ms from cgs (College Park, MD, US)</td>
<td>109.200.218.13</td>
<td>~300km</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.14.228.118</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>168.143.105.162</td>
<td></td>
</tr>
<tr>
<td>Router #4</td>
<td>4ms from cgs (College Park, MD, US)</td>
<td>216.66.14.186</td>
<td>~400km</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.69.219.110</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>195.22.206.99</td>
<td></td>
</tr>
</tbody>
</table>

Range implied by RTT

Range implied by RTT

Range implied by RTT

Range implied by RTT
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

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

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

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

Router #4
- 4ms from cgs (College Park, MD, US)
- 216.66.14.186
- 4.69.219.110
- 195.22.206.99
- ~400km
Substantial work requires accurate router geolocation

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

*iPlane: An Information Plane for Distributed Services*

Uncovering Performance Differences among Backbone ISPs with Netdiff

Nation-State Routing: Censorship, Wiretapping, and BGP

Detecting Traffic Differentiation in Backbone ISPs with NetPolice

A DISTRIBUTED SYSTEM FOR LARGE-SCALE GEOLOCALIZATION OF INTERNET HOSTS

Effective Diagnosis of Routing Disruptions from End Systems

Residential Links Under the Weather

Solar Superstorms: Planning for an Internet Apocalypse

On inferring regional AS topologies

Measuring and Evaluating Large-Scale CDNs

Packet Caches on Routers: The Implications of Universal Redundant Traffic Elimination

Geography and Routing in the Internet

Out of Sight, Not Out of Mind - A User-View on the Criticality of the Submarine Cable Network

Geographic Locality of IP Prefixes

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
- 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

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**

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

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

**Airport codes are unique.**
Challenge: operators use different dictionaries

Also common: operators embed place names.

Challenge: at least 27 populated places named “Washington”
4 named “Ashburn”
**Challenge:** operators use different dictionaries

<table>
<thead>
<tr>
<th>Router #1</th>
<th>xo.iad02.atlascogentco.com</th>
<th>as2828.was14.ip4.gtt.net</th>
<th>te9-2-0d0.cir1.ashburn-vawas.xo.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router #2</td>
<td>vodafone.iad02.atlascogentco.com</td>
<td>ae-0.vodafone.asbnva02.us.bb.gin.ntt.net</td>
<td>zayo.vodafone.er2.iad10.us.zip.zayo.com</td>
</tr>
<tr>
<td>Router #3</td>
<td>usqas1-rt002i.i3d.net</td>
<td>interactive.edge1.washington111.level3.net</td>
<td>ce-0-4-0-2.r05.asbnva02.us.ce.gin.ntt.net</td>
</tr>
<tr>
<td>Router #4</td>
<td>level3-as3356.e0-51.switch2.ash1.he.net</td>
<td>ae-1-3510.edge1.washington111.level3.net</td>
<td>level3.ashburn2.ash.seabone.net</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Chars</th>
<th>Meaning</th>
<th>e.g.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Place</td>
<td>asbn</td>
</tr>
<tr>
<td>2</td>
<td>State/country</td>
<td>va</td>
</tr>
<tr>
<td>5</td>
<td>various</td>
<td></td>
</tr>
</tbody>
</table>

asbnva = Ashburn, VA, US
Challenge: operators use different dictionaries

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

<table>
<thead>
<tr>
<th>Char</th>
<th>Meaning</th>
<th>E.g.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Country</td>
<td>us</td>
</tr>
<tr>
<td>3</td>
<td>Place</td>
<td>qas</td>
</tr>
</tbody>
</table>

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

Router #4
level3-as3356.e0-51.switch2.he.net
  ae-1-3510.edge1.washington111.level3.net
  level3.ashburn2.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.
    - https://www.caida.org/tools/measurement/scamper/

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

Example routers

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

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

- **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**
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)
- Hoipo: (IMC 2019 + 2020)

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

**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)

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.

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

\[ \text{de-cix1.rt.act.fkt.de.retn.net} \]

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)

(Hoiho: Holistic Orthography of Internet Hostname Observations)

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

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

(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

Consider sequences of alphabetical characters that are defined in the input dictionary.
Consider sequences of alphabetical characters that are defined in the input dictionary.
A string is an **apparent geohint** if measured RTTs are consistent with the location implied by the geohint.
Identify apparent geohints with RTT measurements

A string is an **apparent geohint** if measured RTTs are consistent with the location implied by the geohint.
A string is an **apparent geohint** if measured RTTs are consistent with the location implied by the geohint.
Identify apparent geohints with RTT measurements

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)

```
^\.(\[a-z\]{3})\.(\[a-z\]{2})\retn\net$
```
Build Regular Expressions to Extract Apparent Geohints

(see paper for details)

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

Our method includes a plan for each regex: i.e., what each extraction represents.
Build Regular Expressions to Extract Apparent Geohints

\[
\begin{align*}
\text{ae0-2.rt.irx.vie.at.retn.net} & \quad \rightarrow \quad \text{iata: vie, cc: at} \\
\text{ae1-1.rt.irx.vie.at.retn.net} & \quad \rightarrow \quad \text{iata: vie, cc: at} \\
\text{ae1-4.rt.evo.ams.nl.retn.net} & \quad \rightarrow \quad \text{iata: ams, cc: nl} \\
\text{ae0-4.rt.wbx.sea.us.retn.net} & \quad \rightarrow \quad \text{iata: sea, cc: us} \\
\text{ae1-5.rt.irx.cph.dk.retn.net} & \quad \rightarrow \quad \text{iata: cph, cc: dk} \\
\text{ae1-5.rt.srv.mep.nl.retn.net} & \\
\text{de-cix1.rt.act.fkt.de.retn.net} & \\
^+.+(\{[a-z]\})\.(\{[a-z]\})\.
et\net$
\end{align*}
\]

IATA      CC

Vienna, AT
Amsterdam, NL
Seattle, WA, US
Copenhagen, DK

(see paper for details)
Build Regular Expressions to Extract Apparent Geohints

(see paper for details)

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

<table>
<thead>
<tr>
<th>IATA</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>vie</td>
<td>at</td>
</tr>
<tr>
<td>ams</td>
<td>nl</td>
</tr>
<tr>
<td>sea</td>
<td>us</td>
</tr>
<tr>
<td>cph</td>
<td>dk</td>
</tr>
<tr>
<td>mep</td>
<td>nl</td>
</tr>
<tr>
<td>fkt</td>
<td>de</td>
</tr>
</tbody>
</table>

Vienna, AT
Amsterdam, NL
Seattle, WA, US
Copenhagen, DK

???
???
Learn Geohints not in Dictionary

Consider abbreviations of RTT-consistent populated places

<table>
<thead>
<tr>
<th>Place</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>★ Meppel, DR, NL</td>
<td>30,697</td>
</tr>
<tr>
<td>Meppen, DR, NL</td>
<td>305</td>
</tr>
<tr>
<td>Middelkoop, UT, NL</td>
<td>370</td>
</tr>
<tr>
<td>Frankfurt am Main, HE, DE</td>
<td>650,000</td>
</tr>
<tr>
<td>Frankenthal, RP, DE</td>
<td>47,438</td>
</tr>
<tr>
<td>Falkenstein, DE</td>
<td>9,528</td>
</tr>
<tr>
<td>+ 5 other locations</td>
<td></td>
</tr>
</tbody>
</table>

Name of candidate populated place must match first letter in abbreviation. Prefer places with known facilities, then places with higher population. ★ Place has a facility listed in PeeringDB
Validation of learned geohints against ground truth

<table>
<thead>
<tr>
<th>aorta.net</th>
<th>as8218.eu</th>
<th>geant.net</th>
<th>gtt.net</th>
<th>he.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/4</td>
<td>3/3</td>
<td>8/8</td>
<td>12/12</td>
<td>4/4</td>
</tr>
<tr>
<td>(75%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ntt.net</th>
<th>retn.net</th>
<th>seabone.net</th>
<th>tfbnw.net</th>
<th>zayo.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>17/18</td>
<td>25/34</td>
<td>14/15</td>
<td>2/14</td>
<td>4/4</td>
</tr>
<tr>
<td>(94.4%)</td>
<td>(73.5%)</td>
<td>(93.3%)</td>
<td>(14.3%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>aorta.net</th>
<th>as8218.eu</th>
<th>geant.net</th>
<th>gtt.net</th>
<th>he.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/4</td>
<td>3/3</td>
<td>8/8</td>
<td>12/12</td>
<td>4/4</td>
</tr>
<tr>
<td>(75%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ntt.net</th>
<th>retn.net</th>
<th>seabone.net</th>
<th>tfbnw.net</th>
<th>zayo.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>17/18</td>
<td>25/34</td>
<td>14/15</td>
<td>2/14</td>
<td>4/4</td>
</tr>
<tr>
<td>(94.4%)</td>
<td>(73.5%)</td>
<td>(93.3%)</td>
<td>(14.3%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>aorta.net</th>
<th>as8218.eu</th>
<th>geant.net</th>
<th>gtt.net</th>
<th>he.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/4</td>
<td>3/3</td>
<td>8/8</td>
<td>12/12</td>
<td>4/4</td>
<td></td>
</tr>
<tr>
<td>(75%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td>(100%)</td>
<td></td>
</tr>
<tr>
<td>ntt.net</td>
<td>retn.net</td>
<td>seabone.net</td>
<td>tfbnw.net</td>
<td>zayo.net</td>
<td></td>
</tr>
<tr>
<td>17/18</td>
<td>25/34</td>
<td>14/15</td>
<td>2/14</td>
<td>4/4</td>
<td></td>
</tr>
<tr>
<td>(94.4%)</td>
<td>(73.5%)</td>
<td>(93.3%)</td>
<td>(14.3%)</td>
<td>(100%)</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>92/117</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(78.6%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hoiho</td>
<td>94.0%</td>
</tr>
<tr>
<td>HLOC</td>
<td>73.1%</td>
</tr>
<tr>
<td>DRoP</td>
<td>56.6%</td>
</tr>
<tr>
<td>undns</td>
<td>21.8%</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>2.56M</td>
<td>2.57M</td>
<td>559K</td>
<td>525K</td>
</tr>
<tr>
<td>with hostname</td>
<td>1.41M (55.0%)</td>
<td>1.39M (54.1%)</td>
<td>84K (15.1%)</td>
<td>84K (16.0%)</td>
</tr>
<tr>
<td>with apparent geohint</td>
<td>225K (8.8%)</td>
<td>220K (8.5%)</td>
<td>29K (5.3%)</td>
<td>31K (5.8%)</td>
</tr>
<tr>
<td>geolocated</td>
<td>195K (7.6%)</td>
<td>183K (7.1%)</td>
<td>26K (4.7%)</td>
<td>27K (5.2%)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Hostnames:</th>
<th>71.7%</th>
<th>99.3%</th>
<th>75.6%</th>
<th>99.6%</th>
<th>93.0%</th>
<th>95.5%</th>
<th>98.5%</th>
<th>95.5%</th>
<th>84.3%</th>
<th>78.7%</th>
<th>88.6%</th>
<th>90.3%</th>
<th>94.7%</th>
<th>96.7%</th>
<th>94.0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hoiho:</td>
<td>63.3%</td>
<td>79.8%</td>
<td>47.3%</td>
<td>81.1%</td>
<td>63.7%</td>
<td>82.8%</td>
<td>77.2%</td>
<td>68.4%</td>
<td>3.9%</td>
<td>55.7%</td>
<td>78.9%</td>
<td>68.3%</td>
<td>5.3%</td>
<td>77.7%</td>
<td>73.1%</td>
</tr>
<tr>
<td>DRoP:</td>
<td>5.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>8.0%</td>
<td>2.0%</td>
<td>48.6%</td>
<td>60.3%</td>
<td>82.8%</td>
<td>0.0%</td>
<td>79.6%</td>
<td>56.6%</td>
</tr>
<tr>
<td>Undns:</td>
<td>61.7%</td>
<td>0.0%</td>
<td>56.5%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>28.4%</td>
<td>72.1%</td>
<td>7.8%</td>
<td>0.0%</td>
<td>479</td>
<td>1238</td>
<td>4128</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>Hostnames:</td>
<td>60</td>
<td>2403</td>
<td>131</td>
<td>472</td>
<td>270</td>
<td>1697</td>
<td>2121</td>
<td>3397</td>
<td>51</td>
<td>479</td>
<td>1238</td>
<td>4128</td>
<td>19</td>
<td>766</td>
<td>17232</td>
</tr>
</tbody>
</table>

Percentage of hostnames with geohints:
- Hoiho:FP
- Hoiho:TP
- HLOC:FP
- HLOC:TP
- DRoP:FP
- DRoP:TP
- undns:FP
- undns:TP
Selected Related Work: **TBG**

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

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

**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

<table>
<thead>
<tr>
<th>Method</th>
<th>Coverage</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Method</td>
<td>94.0%</td>
<td>95.6%</td>
</tr>
<tr>
<td>HLOC</td>
<td>73.1%</td>
<td>85.1%</td>
</tr>
<tr>
<td>DRoP</td>
<td>56.6%</td>
<td>87.2%</td>
</tr>
<tr>
<td>undns</td>
<td>21.8%</td>
<td>98.3%</td>
</tr>
</tbody>
</table>