The Aleph (ℵ): Decoding Geographic Information from DNS PTR Records Using Large Language Models

KEDAR THIAGARAJAN, Northwestern University, USA
ESTEBAN CARISIMO, Northwestern University, USA
FABIÁN E. BUSTAMANTE, Northwestern University, USA

Geolocating network devices is essential for various research areas. Yet, despite notable advancements, it continues to be one of the most challenging issues for experimentalists. An approach for geolocating that has proved effective is leveraging geolocating hints in PTR records associated with network devices. Extracting and interpreting geo-hints from PTR records is challenging because the labels are primarily intended for human interpretation rather than computational processing. Additionally, a lack of standardization across operators – and even within a single operator, due to factors like rebranding, mergers, and acquisitions – complicates the process. We argue that Large Language Models (LLMs), rather than humans, are better equipped to identify patterns in DNS PTR records, and significantly scale the coverage of tools like Hoiho. We introduce *The Aleph*, an approach and system for network device geolocation that utilizes information embedded in PTR records. *The Aleph* leverages LLMs to classify PTR records, generate regular expressions for these classes, and establish hint-to-location mapping per operator. We present results showing the applicability of using LLMs as a scalable approach to leverage PTR records for infrastructure geolocation.

CCS Concepts: • **Networks → Network performance evaluation**; • **Computing technologies → Artificial intelligence**; • **Computing Methodologies → Machine Learning**.

Additional Key Words and Phrases: Large Language Models (LLMs), Internet Geolocation, DNS PTR Records

ACM Reference Format:
Kedar Thiagarajan, Esteban Carisimo, and Fabián E. Bustamante. 2025. *The Aleph* (ℵ): Decoding Geographic Information from DNS PTR Records Using Large Language Models. *Proc. ACM Netw.* 3, CoNEXT1, Article 7 (March 2025), 20 pages. https://doi.org/10.1145/3709374

> All language is a set of symbols whose use among its speakers assumes a shared past. How, then, can I translate into words the limitless Aleph, which my floundering mind can scarcely encompass?
>
> *Jorge Luis Borges. The Aleph*

1 Introduction

Geolocating network devices is essential for various research areas (e.g., [17, 19, 22, 32, 41]) and internet applications. Despite notable advancements over the course of two decades, it continues to be one of the most challenging issues for network practicioners [35]. While end-host geolocation has advanced significantly due to its commercial value, geolocating infrastructure beyond the edge remains difficult. Techniques commonly used for end-hosts do not always translate well to routers and servers. For instance, while latency-based geolocation can be effective for end-hosts, routers often ignore or rate limit ICMP echo requests [16].

One approach for geolocating infrastructure that has proved effective is leveraging geolocation hints in PTR records associated with network devices. Network operators encode physical location hints in DNS hostname strings of network devices to help with troubleshooting and operation [11] and previous work has shown the potential value of leveraging this information [13, 25, 38, 40]. As early as 1999, GTrace [34] used manually assembled collections of regular expressions (regexes) to extract PTR geolocation hints, an approach later extended by IP2geo [33] with the addition of

Authors' Contact Information: Kedar Thiagarajan, Northwestern University, USA, kedarthiagarajan2028@u.northwestern.edu; Esteban Carisimo, Northwestern University, USA, esteban.carisimo@northwestern.edu; Fabián E. Bustamante, Northwestern University, USA, fabianb@northwestern.edu.

host localization. Most recently, several efforts have tried to automate the task of extracting these location hints [13, 23, 25–27, 38].

Extracting and interpreting geo-hints from PTR records is challenging. For starters, the labels are primarily designed for human interpretation rather than computational processing. In addition, there is a lack of standardization across operators in what geographic information is encoded and how; which leads to the development of ad-hoc approaches. Even within a single operator, legacy infrastructure from rebranding, mergers, and acquisitions results in multiple standards that can take decades to converge. For example, although the merger was executed 20 years ago, AT&T still uses South Bell Corporation Global labels, such as `99-170-164-205.lightspeed.tukrga.sbcglobal.net`. This issue often appears in networks with large geographic spans managed by multiple teams and divisions.

Building on prior work [11, 33, 38] Huffaker et al. [13] tries to automate part of the task by searching for geographic encoding based on a previously populated dictionary of geographic-related strings. More recently, Luckie et al. [25] automatically extract and interpret geo-hints embedded into hostnames using regexes informed by a dictionary that includes strings such as airport codes, city, state and country names, and learn simple deviations from geohints such as prefix (e.g., "ash" for "Ashburn") and partial matches (e.g., "ftcollins" for "Fort Collins").

While highly effective, the coverage of these approaches and the associated tools and datasets is limited primarily due to the challenge of scaling up steps required to confirm geographic inferences. For example, CAIDA's Internet Topology Data Kit (ITDK) [5] infers routers' geolocation combining Hoiho [25], the known location of IXPs and the geolocation database Maxmind. When looking at the example collected between January 30 and February 19 2024 (`itdk-2024-02`), the majority of routers are still geolocated using Maxmind, with Hoiho contributing to locate 6.5% of routers in the spanpshot.

In this paper, we introduce *The Aleph*, a new approach and system for device geolocation that utilizes information embedded in PTR records. *The Aleph* is based on the observation that Large Language Models (LLMs), rather than humans, may be better equipped to identify patterns in DNS PTR records and create extraction rules, offering a path to significantly scale the coverage of tools like Hoiho. It leverages LLMs to (1) classify PTR records into distinct groups based on the structure and potential geographic hints, (2) generate regular expressions for these classes, identifying patterns and consistent naming conventions, and (3) map the identified classifications and regex patterns to geographic locations by linking encoded hints with actual place names.

We make the following key contributions:

- We present *The Aleph*, an implementation of our approach using GPT-4 [30] (§3).
- We apply *The Aleph* to a selected set of Autonomous Systems from transit providers, cloud providers, and access networks (§4), derive the associated regular expressions and geolocation hints, and apply them to our dataset of 1.16 billion PTR records.
- We evaluate the extraction capabilities of *The Aleph* with ground truth from several operators and RTT-based measurements (§5).
- We compare geographic information obtained by *The Aleph* to Hoiho and GeoFeeds on a publicly available Internet topology dataset and report on our findings (§6).

We close with a brief discussion of related work in the space, our approach limitations and some future research directions (§7-9).

2 Background

In this section, we describe the uses of DNS PTR records by operators, the wealth of information they encode and the challenges with extracting it. We then briefly discuss how LLMs may offer a better, more scalable approach to address these challenges.

2.1 Geographic Information Encoded in PTR Records

The availability of network information encoded within DNS PTR records has been known and leveraged by the research community for over two decades, since at least as early as 1999 [34]. PTR records can embed rich information about routers and hosts, from geographic hints (city, nearby airport, or country, or even specific street addresses) to infrastructure details such as backbone connections and peering facilities [40], peering links and the entities on either side of these links, network role (e.g., edge), or even specific access technology, such as DSL, HFC, cable, PPP, or FTTH [21]. Table 1 illustrates part of this wide range using different providers in our dataset.

Table 1. A sample of the range of information available in DNS PTR records.

ASName	ASN	PTR Record	Information	Type
Comcast	7922	be-203-pe11.350ecermak.il.ibone.comcast.net	350 E Cermak, Chicago	Address
Orange	3215	amontsouris-699-1-144-39.w109-216.abo.wanadoo.fr	Montsouris, Paris	Neighborhood
Sprint	1239	ip-70-14-63-1.nsvltn.spcsdns.net	Nashville, Tennessee	City
Virgin Media	5089	brhm-netflix-cdn-16.network.virginmedia.net	Netflix	Peering
PCCW	3491	TELEHOUSE-Te0-0-0-32-2-182.br03.frf05.as3491.net	AS3491	ASN
Hurricane Elec.	6939	e0-1.core3.lon1.he.net	core	Use

2.2 Extraction Tools and their Challenges

Several research efforts have aimed to understand and leverage geolocation information embedded in these PTR records [9, 13, 21, 23, 25–27, 33, 38, 40, 42]. Effectively capturing this information is a complex task. As Table 2 illustrates well, network operators utilize a variety of nomenclatures to label their infrastructure, ranging from standard methods for labeling cities, including IATA, UN/LOCODE, and exact city names, to their own custom conventions. Further complicating data extraction, these encodings frequently lack explicit delimiters, fixed lengths, or positions. This diversity requires deciphering each network's embedding convention and creating the corresponding mappings for each. The encoding patterns vary even within a single operator, due to factors such as legacy infrastructure mergers and acquisition, multiplicity of network teams, among others.

Table 2. A range of embedded geolocation in PTR records. Some records were abbreviated to fit in the table.

ASName	ASN	PTR Record	Geographic Mapping
Sprint	1239	ip-70-14-63-1.nsvltn.spcsdns.net	Nashville, Tennessee
Qwest	207	71-208-140-126.ftmy.qwest.net	Fort Myers, Florida
Verizon	701	pool-*.nrflva.east.verizon.net	Norfolk, Virginia
Claro	4230	*.bva.embratel.net.br	Boa Vista, Brazil (BVA also IATA code for Beauvais, Paris)
Sony	2517	*.kngwnt01.ap.so-net.ne.jp	Kanegawa, Japan
China Telecom	4134	*.fz.fj.dynamic.163data.com.cn	Fuzhou, China (FZ is not a standard LOCODE)
Comcast	7922	*northlake.il.ndcchgo.comcast.net	North Lake, Chicago, US

Hoiho [25], the state-of-the-art tool for extracting embedded geographic information from these records, builds on DNS PTR records observed in traceroutes used to construct the CAIDA ITDK dataset. It uses two main building blocks: (1) a dictionary of geographic hints, such as IATA airport codes, city names from a publicly available geographic database GeoNames, LOCODEs, and CLLI,

and (2) a set of regular expressions (regexes) to extract geo-hints, which it refines using natural language processing (NLP). However, as we show in Secs.4-6, many PTR encoded records rely on unconventional or mistaken abbreviations and operator or region-specific encoding and location hints, requiring techniques that can leverage contextual clues to disambiguate their meaning.

2.3 An Opportunity for Large Language Models

Identifying geo-hints embedded in PTR records falls within the scope of Named Entity Recognition (NER) in NLP, a technique used to identify and classify key information (entities) into predefined categories. By applying NER, we can extract and categorize these geo-hints, leveraging NLP's ability to parse and interpret textual data. More specifically, this task is a form of Information Extraction (IE) that converts unstructured text into structured data. Additionally, large language models (LLMs) can help lift language barriers [12, 20], enhancing the understanding and processing of multilingual data, e.g., NTT labeled London as Londen (Dutch).

Recent advancements in Few-Shot Learning (FSL) approaches using LLMs [4] suggest the value of this technique to address our problem. Brown et al. [4] seminal paper shows that zero-, one-, and few-shot settings may at times surpass state-of-the-art fine-tuned models. Zero-shot learning opens new opportunities for enhancing IE, particularly in our domain. Previously, IE methods depended heavily on human-annotated data, yet their performance diminished with each new annotation schema, making manual annotation for each domain impractical. Zero-shot IE systems now employ LLMs to utilize pre-trained knowledge for annotations [37], obtained as a by-product of the pre-training process. We leverage this inherent model knowledge to develop a system that employs modern LLMs to generate patterns from sample records and create extraction rules.

3 *The Aleph*: Approach and System Design

In the following paragraphs, we introduce *The Aleph*, a new approach and system for network device geolocation (Fig. 1). *The Aleph* builds on FSL [4] to create pipelines that take advantage of modern LLMs' NER capabilities to learn example patterns and generate extraction rules from a few instances. *The Aleph* is implemented using OpenAI's GPT-4 Turbo [30] with `temperature` set to 0 and `Top P` probability mass to 1. The `temperature` setting is used to ensure that the LLM always outputs the most likely next token, leading to output that should be reproducible unless the model weights are changed.

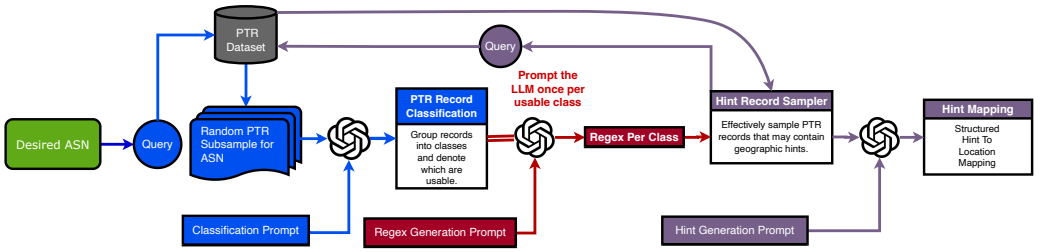


Fig. 1. A visual representation of the steps taken by *The Aleph* to decode DNS PTR records. For each AS, we use PTR records from OpenIntel's daily scan of the ARPA zone database to group records into classes (§ 3.3), generate regular expressions per class (§ 3.4), and extract geo-hints (§ 3.5) from a sample of PTR records.

We describe the pipeline architecture and prompting strategies in *The Aleph*, using examples of PTR records from AT&T-AS7018, a provider with diverse encoding conventions, for illustration. Listing 1 includes a handful of these DNS PTR records.

```
107-194-77-243.lightspeed.austtx.sbcglobal.net     usr38.clearsail.net
76-225-54-166.lightspeed.bkfdca.sbcglobal.net      adsl-69-219-222-66.dsl.chcgil.ameritech.net
adsl-99-36-211-162.dsl.skt2ca.sbcglobal.net        adsl-70-254-151-53.dsl.wcfltx.swbell.net
adsl-074-236-001-094.sip.mia.bellsouth.net         20.suh78.dlls.dlstx31ur.dsl.att.net
```

Listing 1. Example of PTR records for AT&T (AS7018).

3.1 Problem Statement and Processing Separation

The Aleph's goal is to create regular expressions that extract geographic information from PTR records and output this geographic data in a detailed, structured format.

We initially explore a one-shot approach to this problem, using the same LLM and hyperparameters[1] as for *The Aleph*. The one-shot approach attempts to generate both regexes and hint mappings to extract geographic hints from DNS PTR records.

Listing 2 displays the output when applied to our case study of AT&T, including a set of regular expressions and a map of hints to locations (e.g., 'austtx -> Austin, Texas'). Several issues arise with the generated regexes, leading to incorrect or inadequate results. While the mapping of hints to locations appears accurate, the regexes themselves present a rigid pattern, searching for a three-letter city abbreviation followed by a two-letter state code (e.g., `tx`, `ca`). This approach fails to accommodate city abbreviations extending beyond three letters, like `aust`, and does not effectively capture the variety of geographic hints in PTR records. For example, these regexes would not recognize longer or non-standard abbreviations such as `dlls` for Dallas or city abbreviations not followed by a two-letter state code (e.g., `mia` without `fl`).

```
# prompt
You are an expert in Computer Networks and have the following tasks: First generate a list of
regular expressions to extract geographic hints from all types of records within the provided
records. Then create a map of hints to locations, which can be cities identified by
abbreviations, full names, or iata codes, states, and countries from the following record in
JSON. Do not output anything else.

# outputs
Regular Expressions:
1. `(?i)\b([a-z]{3})tx\b` - Extracts three-letter abbreviations ending with 'tx' (Texas).
2. `(?i)\b([a-z]{3})ca\b` - Extracts three-letter abbreviations ending with 'ca' (California).
3. `(?i)\b([a-z]{3})il\b` - Extracts three-letter abbreviations ending with 'il' (Illinois).
4. `(?i)\b([a-z]{3})fl\b` - Extracts three-letter abbreviations ending with 'fl' (Florida).

Map of Hints to Locations:
- `austtx` -> Austin, Texas
- `bkfdca` -> Bakersfield, California
- `chcgil` -> Chicago, Illinois
- `skt2ca` -> Stockton, California
- `wcfltx` -> Wichita Falls, Texas
- `mia` -> Miami, Florida
- `dlls` -> Dallas, Texas
```

Listing 2. Example of the regexes and hint mappings generated for AT&T (AS7018) in a one-shot approach.

Instead of using a one-shot approach, *The Aleph* divides the process of generating regular expressions and geohints into intermediate steps to improve precision. By mitigating overfitting using providing illustrative examples, few-shot learning significantly enhances the accuracy and

[1]Parameters that influence the learning process, such Top P and temperature. Top P is set at 1 to make sure all possible tokens are considered, and temperature is set to 0 to ensure the most likely token is always chosen.

robustness of LLM outputs compared to one-shot methods – particularly for tasks requiring deeper contextual understanding, nuanced pattern recognition, and robust generalization [4].

3.2 Generating Geohint Extraction Rules

The Aleph's approach for generating extraction rules consists of three stages: (1) creating groupings to showcase examples from each class, (2) generating regexes based on these examples, and (3) creating generalizable hint mappings. *The Aleph* conducts a per-network rules generation and hints inferences, assuming each network operator employs a unique set of encoding patterns and naming conventions with minimal overlap with those of others. To enable reproducibility, we have made all prompts, PTR records, and intermediate outputs from each stage of the process publicly available at https://thealeph.ai/demo.

3.3 Encoding Pattern Categorization

The first stage of *The Aleph* (in blue in Fig. 1) separates all PTR records of a given network into distinct categories based on their encoding patterns. This classification into categories is required as LLMs face challenges to identify PTR encoding structures and, consequently, generate regexes for them when presented with multiple PTR records using different encoding patterns (see §3.1),

To guide LLMs in this stage of classifying PTR records into categories, the prompt instructs the LLM to group PTR records based on similar encoding patterns, for example, records embedding geographic and operational information.

As LLMs can only process a limited number of tokens in their context, *The Aleph* restricts the number of PTR record examples used for classification to GPT-4 Turbo's maximum context length [29], which in our implementation varies between 337 and 642 examples.

Table 3 illustrates this, showing the seven categories assigned to AT&T-AS7018 in the example from Listing 1, along with a representative PTR record for each category.

Table 3. Example PTR records per class for AT&T AS7018.

Class	Examples
sbcglobal	99-170-164-205.lightspeed.tukrga.sbcglobal.net
ameritech	67-37-109-75.ded.ameritech.net
mycingular	mobile-166-199-079-114.mycingular.net
swbell	adsl-70-254-151-53.dsl.wcfltx.swbell.net
bellsouth	adsl-074-236-001-094.sip.mia.bellsouth.net
att	20.suh78.dlls.dlstx31ur.dsl.att.net

3.4 Extraction Rules Generation

After separating PTR records into distinct categories, the next step involves obtaining a `regex` to extract geo-hints (shown in red in Fig. 1).

To generate these extraction rules, *The Aleph* provides up to five PTR record examples per category[2] identified in the previous step (§3.3). In addition to passing these examples, *The Aleph* instructs the LLM with a tailored FSL prompt designed to produce a `regex` that extracts geo-hints specific to each PTR record class for a given network provider. Listing 3 displays the resulting regexes for each class identified in the AT&T-AS7018 example.

```
1  # patterns for AT&T
2  1. (?<=lightspeed\.)[a-z]+[a-z]{2}(?=\.sbcglobal\.net)
```

[2]The number of examples is limited by GPT-4's maximum context length

```
2. dsl\.([a-z0-9]+)\.[a-z]{2}
3. ([a-z0-9]+)\.([a-z]+[a-z]{2})
4. sip\.(\w{3})\.bellsouth\.net
5. (?<=dsl\.)[a-z]+[a-z]{2}(?=\.pacbell\.net|\.ameritech\.net)
6. dsl\.([a-z]+[a-z]{2})\.ameritech\.net
```

Listing 3. Step 2 Regexes

This set of regexes, compared to those generated by the one-shot approach (see Lst. 2), more effectively captures the structure and common patterns within AT&T's PTR records. For example, the use of positive look-behind and look-ahead assertions in patterns such as `(?<=lightspeed\.)[a-z]+[a-z]{2}(?=\.sbcglobal\.net)` ensures matches are only made when the specific context is present. These regexes are also flexible to capture variations in subdomain structures and are specifically designed to encompass a wider array of geographical hints. Patterns such as `dsl\.([a-z0-9]+)\.[a-z]{2}`, which allow for alphanumeric city abbreviations followed by state codes, and `sip\.(\w{3})\.bellsouth\.net`, which targets three-character hints within Bell South domains, display this capability.

3.5 Deciphering Geo-hints: Creating a Geo-hint-to-location Database

In addition to the LLM-generated extraction rules, *The Aleph* integrates a geohint-to-location mapping database, which has also been populated through a prompt-based inference, using the pipeline shown in purple in Fig 1.

To extract geo-hints, *The Aleph* begins by applying the initial steps – encoding pattern categorization (§3.3) and extraction rule generation (§3.4) – to a randomly selected subset of PTR records from a given network.

Once these extraction rules are created, *The Aleph* applies them to a different set of PTR records from the same network to retrieve geo-hint samples.

Next, *The Aleph* leverages an LLM to map these geo-hints to geographic locations, constructing a comprehensive geo-hint-to-location database. Given the diversity in naming conventions used by network operators – such as IATA codes, UN/LOCODEs, or custom labels – *The Aleph*'s database effectively captures and manages this variability.

Table 4. Step 3: Locations Combined

	skt2ca	**chcgil**	**mia**	**wcfltx**	**austtx**	**bkfdca**	**dlstx**
City	Stockton	Chicago	Miami	Wichita Falls	Austin	Bakersfield	Dallas
State	CA	IL	FL	TX	TX	CA	TX
Country	US	US	US	US	US	US	US

To improve coverage, *The Aleph* implements an iterative process that refines geo-hint extraction through resampling as more PTR records are analyzed. Unlike the earlier one-shot approach (Lst. 2). Table 4 shows the precision of *The Aleph*'s mappings. Examples like 'skt2ca' for Stockton, and 'dlstx' for Dallas, captured by regexes (2) and (3) illustrate how regex guidance ensures that the extracted hints match regex capturing groups.

4 Generating rules with *The Aleph*

We used a snapshot of the DNS PTR records collected by OpenIntel [31] in February 2024, and a subset of ASes selected for coverage to generate regular expressions and hint mappings with *The Aleph*. In the next paragraphs we describe the selected ASes, including our criteria for their

inclusion. We close with an analysis of the generations of *The Aleph* (Sec. 4.2), and discuss the complexity and scale of the problem of extracting geographic information from PTR records.

4.1 Selecting Evaluation Cases

DNS PTR records map IP addresses to domain names for reverse DNS lookups. Managed by IANA within the ARPA zone, these records are configured by network operators to reflect the domain assignments for their allocated IP addresses [15, 36]. OpenIntel conducts daily scans of the ARPA Zone database and maintains a repository of daily snapshots of all PTR records and their operators; we use a subset of these operators in our analysis.

Table 5. Description of OpenIntel Dataset and subset used for training *The Aleph*

	Total	Training
# ASes	51,840	2,646
Transit	151	151
Content	-	15
% Eyeballs	93.33	84.31
# PTR records	1,282,817,253	1,238,198
% PTR records	100	0.01

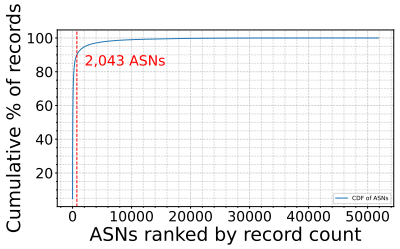


Fig. 2. Percentage of PTR Records managed by subsets of networks in OpenIntel Dataset. ≈4% of AS manage over 90% of the records.

We selected a subset of 2,646 networks to ensure comprehensive coverage of DNS PTR records from the OpenIntel database, enhance geographic representation, and capture a significant portion of the global Internet population. Of these, we chose 2,043 networks by ranking all networks according to their share of PTR records and selecting those that collectively accounted for 90% of all records. As Fig. 2 shows, the distribution of PTR records managed per networks is heavy-tailed; extending coverage to 100% would require including 49,193 more networks.

To further improve geographic diversity and Internet population coverage, we included 603 additional networks as follows:

Access Networks. We use APNIC's Internet Population report [14] of October 5, 2023 to identify 390 ASes that together cover 80% of global Internet population. For geographic diversity, we include the largest AS by user population for each country, totaling 171 additional ASes in 165 countries.

Transit Networks. We employ CAIDA's AS-relationships [24] to identify large transit networks that do not rely on any upstream provider, including Lumen (AS3356), Arelion (AS1299), Comcast (AS7922), and Orange (AS3215), which collectively manage 16,965,810 PTR records. The typical large geographic coverage of these networks means they are likely to benefit from labeling their infrastructure. Indeed, many of these were already included in the initial PTR coverage selection.

Content Providers. We also included the networks of the 15 most prominent content providers, as defined by Bottger et al. [3] and Carisimo et al. [8]. This set includes: Apple Inc (AS714), Amazon.com (AS16509), Facebook (AS32934), Google Inc. (AS15169), Akamai Technologies (AS20940), Yahoo! (AS10310), Netflix (AS2906), Hurricane Electric (AS6939), OVH (AS16276), Limelight Networks Global (AS22822), Microsoft (AS8075), Twitter, Inc. (AS13414), Twitch (AS46489), Cloudflare (AS13335), Verizon Digital Media Services (AS15133). Their networks also cover large geographic areas and are thus likely to embed geohints in their PTR records.

This combined dataset represents over 84% of the Internet population and manages 90% (1,164,978,231) of all PTR records available in the OpenIntel database. The characteristics of the OpenIntel data, and the specific slice used for our analysis are detailed in Table 5.

4.2 Regex Generation

We use *The Aleph* to generate regular expressions and geolocation hints for the 1.16 billion records from the selected set of 2,646 ASes. From the total set of records, we were able to extract geographic information from 224,172,222 (19%) records collectively managed by 1,551 operators (58% of our total set). The application of *The Aleph* to this dataset generated 4,910 unique regular expressions after approximately 2 days (individual accounts are rate-limited; the time covers the period from issuing the first request to receiving the final response) at a total cost of $500 USD using the OpenAI GPT-4 Turbo Model, and yielded 16,108 geo-hints spread across 6,025 distinct locations in 206 different countries. Table 6 presents a summary of these results.

Table 6. Summary of the results from applying *The Aleph* to the selected 2,646 ASes.

Mapping results	
# of ASNs	2,646
Frac. of Eyeballs	84.31 %
Frac. of PTR Records	90%
# of regexes	4,910
# of Hints	16,108
# of Countries	206
# of Cities	6,025
% of Records encoding city-level data	19%
% of Operators encoding geographic data	58%
Cost ($)	$500
Runtime	≈ 2 days

Table 7. ASes with the largest numbers of unique regular expressions.

Provider	ASN	# of Classes
Multnomah Education District	32522	15
Orange - Cote d'Ivoire	29571	12
Verizon	701	10
Gabon Telecom	16058	10
Sudatel	15706	10
Telecom Algeria	36947	10
Cogent	174	9
AT&T	7018	7
Amazon	16509	7
Google	15169	6

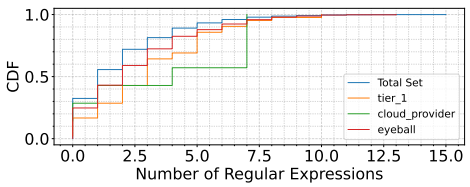


Fig. 3. Number of regular expressions generated per provider in our dataset.

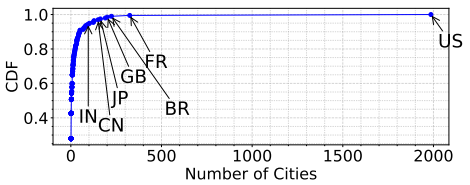


Fig. 4. Distribution of unique city hints found for each country in our dataset.

Encoding-Pattern Diversity. Figure 3 shows the CDF of the number of regular expressions used by each of networks in our dataset, along with a breakdown by network type. Of all networks, 50% use 3 or more classes, while 20% use more than 5. The network with the highest count of unique regular expressions is Multnomah Education District in Oregon, which has one regular expression per city within its coverage area. Orange Côte d'Ivoire follows with 12 regular expressions, likely due to the variety of services it offers under distinct second-level domains (e.g., aviso, vipnet) and associated subdomains.

Cloud providers generally have fewer regex classes, with Amazon and Google as exceptions likely due to their multiple services (e.g., EC2, Cloudfront, and S3). Most other cloud providers use

Table 8. Cities with multiple geo-hints

City	#	Examples
Tokyo	123	tok, tokyjp09.jp, ap-northeast-1, tyo, tkyojpn1, tky, tokyff, tv2-tokyo, tokyjp08.jp, Gtokyo, Itokyo, tokynt, tokyjp05.jp, tokyo, tkyc, hnd
Chicago	95	ord1, chi-il, chcg, chi1, CHGO, chil, chic, ord12, chi-stk, ord, chi, chcgil, chcoilx1, ch, ord13, eqch, ord0, ord3
Paris	74	paris, th2-1, seg75, vau75, dc3-1, pa3-1, montsouris, prs, parscmta, eu-west-3, grtpare, dc3-2, cdg, les01, par, pvu-paris
Los Angeles	67	lax2, la1, laxg, lax-ca, lsancax1, LAX1, lax, lsan03, losa2, lsan, losangelesheredia, lvw-losangeles, los-angeles, lax08, lax3, lax0, lax04, lax4, los, la, losa, lax1
London	65	london, lon7.uk, uklond, lndngbr1, lon2, londresLondon, lnd, west-lon, ldnst, lhr, LDN2, lon10, east-lon, ldy, uk-lon1, grtlon, ucl, uklon, lon, harleystreetmd, londres-wandsworth, lon1, lon3.uk, lrnlon, londonah, ldnbt, lhn, lon.uk, soho, londres-oxfordstreet1, london.on, ldn eu-west-2, lgw, ldncng, lon3, l78-london, lhr1, knightsbridge, lhr6
New York	54	newy, lga12, lga7, jfk, nyy-new-york, ny, n75-newyork, lga6, ny325, jfk04, lga1, lga2, lga, jfk02, newy2, lga3, nyc, nynyc, jfk01, new, nycmny, lga11, nwyynyx1, newy32aoa, nymnny, nyk

one or two regex classes, mapping regional or datacenter names to city locations. *The Aleph* is able to directly map these hints to city-level locations.

In ISP networks, some large U.S.-based providers (e.g., AT&T-AS7018, Qwest-AS209, Cogent-AS174, and T-Mobile-AS1239) show a high number of regex classes, likely due to mergers and acquisitions, while some networks (e.g., Verizon and Comcast) use specific naming conventions to indicate network types or device purposes (e.g., "fios" for Verizon or "hfc" for Comcast).

Table 7 presents the ten networks above the 80^{th} percentile in number of unique regular expressions, including Multnomah Education District, Amazon, AT&T and Cogent. Amazon uses a variety of encoding patterns across its different datacenter regions and services, while the diversity in encoding at AT&T, Verizon, and Cogent can be explained by their extensive geographic spans and legacy infrastructures. The case of the African operators seem to follow a different model. All these network employ unique patterns tailored to different service types, such as educational, residential, government, and enterprise sectors. We include detailed examples of the encoding patterns used by these networks in Appendix C.

Geographic prevalence. When examining the encoding of city-level geographic information by operators across various countries and regions, we find a single labeled city in 40% of countries, with the median value of 2 cities per country. Looking at the tail of the distribution, the leading countries in number of labeled cities in PTR records include the United States (1,989 cities), France (325 cities), Brazil (224 cities), Japan (160 cities), China (150 cities), and India (96 cities). This group, unsurprisingly, includes large, populous countries like the United States (331 million people, 9.8 million km^2), China (1.4 billion people, 9.6 million km^2), Brazil (213 million people, 8.5 million km^2), and India (1.38 billion people, 3.3 million km^2) – four of the world's seven largest countries. Additionally, Japan, France, and Great Britain, though smaller in size, are densely populated, have extensive network infrastructure, and host international interconnection points with various submarine cable networks touching their shores.

Geo-Hint Diversity. Our next analysis examines the complexity of inferring city locations from diverse geo-hints. Overall, we find that most operators rely on custom labels. Table 9 categorizes geo-hints into seven types, including standards like IATA and ICAO codes, United Nations conventions (UN/LOCODE), and custom labels unique to each provider. While among standardized methods, IATA airport codes (e.g., `jfk` and `ord`) are the most common, nearly 66% of the extracted geo-hints are custom hints. For example, Arelion-AS1299 (formerly Telia) uses labels like `nyk` for New York City and `ffm` for Frankfurt.

The diversity of geo-hints across cities – both among different operators and even within a single operator – poses a significant challenge, as the sheer number of geo-hints can be quite large. This complexity places a substantial burden on geohint-to-location mapping generation methods, which must interpret an array of custom naming conventions. However, *The Aleph* demonstrates that LLMs offer a suitable alternative for generating these mappings, even for highly custom geo-hints.

Figure 5 shows the cumulative distribution of geo-hints per provider for individual cities. While 90% of networks use a single geo-hint per city, 10% apply multiple hints, producing a long-tailed distribution. Cities with numerous unique hints often follow patterns seen in Ashgabat, labeled by State Company of Electro Communications Turkmenistan-AS20661, using combinations of the city's name, abbreviations, and prominent businesses. Similarly, AS17882 - Univision Mongolia labels Ulaanbaatar using various business-related identifiers. University and educational networks frequently demonstrate this encoding diversity as they embed building or institution names within PTR records. For instance, Renaeter-AS2200, a French educational network, labels Paris with hints like `univ-paris`, `u-paris`, `u-paris-est`, `u-paris-assas`, and `ipgp`.

Figure 6 displays the cumulative distribution of geo-hints employed for each city within our dataset. While most cities have a single geo-hint, the tail of the distribution reveals high variation in larger cities. For example, Tokyo, Chicago, Los Angeles, and New York City are labeled with as many as 123, 95, 67, and 54 unique geo-hints, respectively, as shown in Tab. 8. Detailed examples of geo-hints for the 20 most frequently labeled cities are provided in Appendix D.

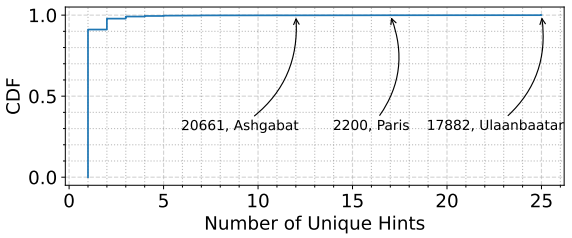


Fig. 5. Distribution of unique encodings per provider per city.

Table 9. Types of Geographic Hints Extracted

Type of Hint	Count	%
Custom	13,438	65.69%
IATA Code	3,413	16.68%
Place	2,467	12.06%
Country	994	4.85%
ICAO Code	54	0.26%
LOCODE	71	0.34%
Facility	19	0.09%

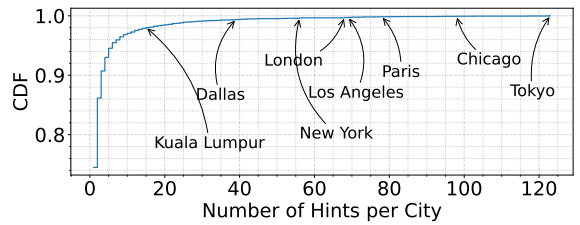


Fig. 6. Distribution of the number of hints per city.

5 Evaluating *The Aleph*

In this section, we present an evaluation of *The Aleph* by (1) comparing results against ground truth data provided by network operators (§5.1), and (2) validating inferred locations using RTT-based active probing (§5.2).

5.1 Ground Truth Validation

Our ground-truth validation includes 3 networks of varying types and sizes, each with a broad geographic footprint and diverse geo-hints in their PTR records. The validation set consists of an access network (COMCAST-AS7922, the largest eyeball network in the US[3]), a transit network

[3]APNIC, October 28, 2024: https://stats.labs.apnic.net/cgi-bin/aspop?c=US&d=25/10/2024

(Arelion-AS1299[4]), and a Japanese ISP (Internet Initiative Japan, IIJ-AS2497). For COMCAST and IIJ we rely on data provided by the operators (COMCAST, IIJ). For Arelion, we leverage public data on their Looking Glass (LG) service, which includes city-to-geo-hint mappings.

COMCAST-AS7922. We validated *The Aleph* mappings for COMCAST's PTR records at the city and state levels within the US using ground truth data comprising IP addresses mapped to latitude, longitude, and bounding radii. COMCAST's ground-truth dataset contains 24,465,865 IP addresses belonging to 3,265 prefixes appearing in the OpenIntel snapshot. This dataset confirmed that *The Aleph* correctly inferred the location of 99.3% IP addresses with PTR records with city-level granularity (210,926 IPs) and 100% for those only having state-level granularity. The remaining 0.7% that were not correctly inferred at the city level correspond to mislabeled records, for instance, a PTR record with a geo-hint pointing to San Francisco, although the IP address was actually assigned to a device in Denver.

IIJ-AS2497. To validate *The Aleph*'s PTR-to-location inferences for IIJ, we exchanged our inferences with IIJ's operators, allowing them to verify the accuracy of our mappings. *The Aleph* identifies IIJ's presence in 31 distinct locations associated with 31 PTR records, of which 30 inferences were confirmed as accurate. The one case where *The Aleph* produced an incorrect mapping was when interpreting the geo-hint `mtk` from records like `mtk001sagnw00.IIJ.Net`. *The Aleph* interpreted this as being in Matsukawa, instead of Mitaka as reported by IIJ. This is due to the known issue of ambiguity in three-letter abbreviations for Japanese city names.

Arelion-AS1299. For Arelion – formerly Telia – we relied on publicly available data from the company's LG servers[5], where each server location is specified by both Arelion geo-hints and the actual city name. We scraped these hints from the LG website and compared them to the locations inferred by *The Aleph. The Aleph* extracted 47 geo-hints, 38 of which overlapped with 95 total hint mappings contained in the LG data, achieving 100% accuracy for the overlapping locations.

5.2 Enhancing Confidence in Inferred Locations with RTT-based Active Probing

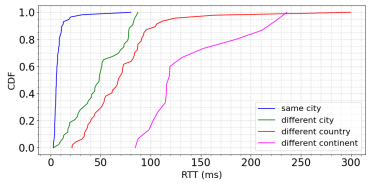
We use active probing to enhance confidence in the inferred locations. By sending probes from vantage points with known and reliable locations, we verify whether the inferred IP address location aligns closely with the vantage point. We issue probes from across a selection of access, transit, and content provider networks, leveraging RIPE Atlas nodes distributed across diverse regions.

Table 10. List of AS contained within the active probing validation set.

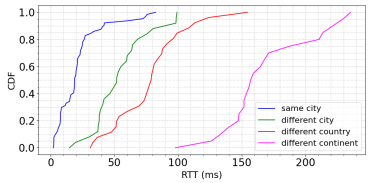
Type	AS Name-ASN
Access	Verizon-AS701 (NA), AT&T-AS7018 (NA), Claro-AS4230 (LAC), Bolivia Telecom-AS27882 (LAC), NTT-AS4713 (EAP), China Telecom-AS4134 (EAP), Orange-AS3215 (EAC), NTL-AS5089 (EAC), Du-AS15802 (MENA), STC-AS25019 (MENA), BSNL Backbone-AS9829 (SA), Airtel Broadband-AS24560 (SA), VODACOM-AS29975 (SSA), BTC-GATE-AS14988 (SSA)
Transit	GTT-AS3257, Orange Transit-AS5511, NTT-AS2914, Internet2-AS11164, Internet2-AS11537, Sprint-AS1239, Cogent-AS174, Qwest-AS209, TierPoint-AS30340, PCCW Global-AS3491, Level3-AS3356, OpenTransit-AS5511, TATA Communications-AS6453, Liberty Global-AS6830
Content and Cloud	Google-AS15169, Salesforce-AS14340, Netflix-AS2906

[4]Ranked second in CAIDA's AS-RANK, October 2024:https://asrank.caida.org
[5]Arelion' s Looking Glasses: https://lg.twelve99.net

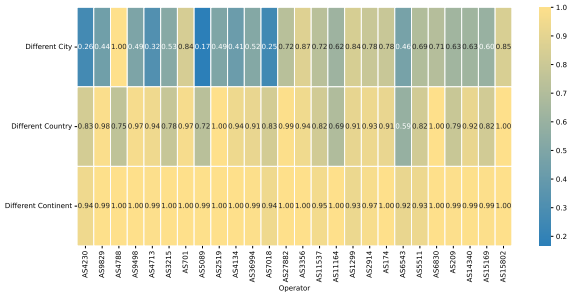
Selecting Networks for Analysis. Our network selection criteria for this analysis depend on the type of network. For access networks, we divide the world into seven regions based on the World Bank's classification [1] and select the two networks with the largest eyeball populations in each region. For content and transit networks, we randomly choose 12 networks from these categories within our dataset. The selection process is iterative, ensuring the inclusion of networks that: (1) encode geographic information in PTR records, and (2) have routers responsive to active probing. Table 10 lists all networks included in our validation.



(a) Verizon distribution.



(b) AT&T distribution.



(c) Heatmap of KS scores.

Fig. 7. Distributions of RTT data for AT&T and Verizon and a heatmap of KS scores for measurements conducted across different categories. Probes were selected in a different city within the same country, a different country within the same continent, or a different continent.

Measurement Results. Figure 7 presents two examples of results from our measurement campaigns on the left side. Appendix E includes a more extensive set. To compare our findings across operators, we rely on Kolmogorov-Smirnov (KS) distance between the distribution of RTT measurements. The KS test quantifies the distance between two empirical distributions (e.g., a sample and reference distribution) and assigns a score of 0 when both are drawn from the same distribution and a score of 1 when random variables with disjoint supports generate them.

We first gather RTT data for 'same city' pairs – where the vantage point and the probed IP address are expected to be co-located – and use it as the reference distribution. We calculate the KS distances between the reference RTT distribution and the RTT distributions for different cities, countries, and continents. All KS distance results in our analysis demonstrated statistical significance (p-values < 0.05), confirming that the observed differences are unlikely to be due to chance.

We aggregate the results across all providers and present them as a heatmap of the calculated KS distances in Fig. 7. Our findings support the initial hypothesis: as probes originate farther from the city indicated by the geo-hint, the KS distance increases, strengthening the confidence in our inferences. These results are influenced, in part, by the characteristics of the networks and vantage point deployments. For example, NTL and Virgin Media in the UK exhibit smaller differences compared to other eyeball ISPs, likely due to the UK's compact geographic size and more localized network structure.

6 *The Aleph* and other Geolocation Methods

In this section, we utilize geolocation information obtained by *The Aleph* and Hoiho – widely recognized as the state-of-the-art for PTR-to-location mappings (see §8) – to analyze the results each method can extract from a given dataset. Additionally, we compare *The Aleph* with Geofeeds, a fundamentally different system that uses operator-published prefix-to-location mappings to serve a similar role from the user's perspective. The latter comparison is included in Appendix B.

6.1 Analysis' Approach

Our analysis uses PTR records from CAIDA's ITDK snapshot for February 2024 (`itdk-2024-02`). These PTR records were gathered by mapping IP addresses observed in CAIDA's Archipelago (Ark) platform [6] daily, network-wide traceroute campaigns conducted between January 30 and February 19, 2024. In addition to PTR records, the ITDK provides alias resolutions (IP-to-router mappings) for addresses appearing in these traceroutes.

The dataset comprises 138,067,845 IPv4 addresses announced by 71,208 ASes and mapped to 3,584,811 routers. Of these routers, 1,936,691 (54%) have at least one IP address with a non-empty PTR record. We leverage *The Aleph*'s regular expressions and hint mappings alongside Hoiho's inferences and GeoFeeds data. The GeoFeeds dataset includes prefixes advertised by 3,356 networks collected with `geofeed-finder` [7], while Hoiho's regular expressions are derived specifically from the PTR records within this dataset [25].

6.2 *The Aleph* and Hoiho

We use *The Aleph* and Hoiho to analyze different networks, focusing on regular expressions and geohint-to-location mappings. We select seven networks that vary in purpose, size, and geographic footprint: BSNL-AS9829, ChinaTelecom-AS4134, NTT-AS2914, Arelion-AS1299, Claro Brazil-AS4230, AT&T-AS7018, and Qwest-AS209. While this analysis is not exhaustive, we leave a more comprehensive evaluation of both methodologies for future work.

Method. Our geolocation method involves identifying all IP addresses and their PTR records in the ITDK dataset, specifically from `itdk-2024-02`, announced by each provider. We utilize *The Aleph* to develop regular expressions and geohint-to-location mappings, comparing our results with Hoiho's established mappings from CAIDA's website [28]. Our study covers 5,869,676 IP addresses, 5,299,691 (90%) of which have associated PTR records totaling 107,311 unique PTR records. Retraining Hoiho with new PTR records and operators is possible but beyond this study's scope; instead, our choice was to train *The Aleph* with PTR records from ITDK to compare *The Aleph* and Hoiho performance when trained on the same dataset.

Table 11. Comparison of Regexes and Hints for Various Networks

Network		Regexes		Hints Found in ITDK		Hints With Locations		Unique Locations	
ASName	ASN	Hoiho	*The Aleph*	Hoiho	*The Aleph*	Hoiho	*The Aleph*	Hoiho	*The Aleph*
Qwest	209	1	4	71	91	71	91	71	77
AT&T	7018	3	6	261	272	29	272	29	241
Claro BR	4230	1	3	10	51	10	51	10	51
Arelion	1299	2	2	9	58	9	58	9	55
NTT	2914	2	5	91	97	40	97	39	81
BSNL	9829	1	4	3	11	3	11	3	11
China Telecom	4134	0	4	0	77	0	77	0	77

General Observations. Table 11 presents the results of both approaches across all networks, detailing the number of regular expressions generated, geo-hints found in ITDK, geo-hints successfully mapped to locations, and the total count of unique locations.

We observe an increase in the number of generated regular expressions and geo-hints mapped to a location when comparing the results of *The Aleph* with Hoiho. This difference arises, in part, because CAIDA is not allowed to make CLLI locations publicly available due to licensing constraints, while the LLM-based approach offers an open alternative that effectively fills this gap.

False Positives and False Negatives. We examine whether *The Aleph* can improve accuracy by resolving challenging inferences. Specifically, Hoiho's dataset includes both false positives (a regex extracted a geo-hint that violates speed-of-light constraints in Hoiho's validation dataset) and false negatives (a regex failed to map a geo-hint despite Hoiho detecting one).

We find that *The Aleph* avoids certain false positives and false negatives for both Qwest and Claro Brazil. For instance, Qwest uses a non-standard CLLI code (`phnx` instead of the more complete `phnxaz`), which *The Aleph* successfully maps. In the case of Claro Brazil, Hoiho detects 10 unique locations while *The Aleph* identifies 51. Of those 51, 18 and 3 correspond to false negatives and false positives in Hoiho's dataset, respectively. The discrepancies arise from ambiguous three-letter local abbreviations, such as `bva` for Boa Vista and `ntl` for Natal, which Hoiho mapped to Beauvais Airport in Paris and Williamtown in Australia, respectively, due to exact IATA matches in its geo-dictionary.

Additional Regional Encodings. As in the case of Claro Brazil, some providers rely on unique encodings. For instance, China Telecom-AS4134 uses non-standard two-letter city abbreviations (e.g., `fz` for Fuzhou and `qz` for Quanzhou). Another example is BSNL-AS9829, where Hoiho detects only three hints based on exact string matches for city names, while *The Aleph* disambiguates three-letter abbreviations like `kol`, `hyd`, and `mum` (Kolkata, Hyderabad, and Mumbai).

Take-aways. This analysis highlights the strengths of The Aleph's LLM-based approach in geolocation tasks, particularly its ability to generate more comprehensive regular expressions and map a greater number of geo-hints to unique locations compared to Hoiho. By addressing challenges such as ambiguous encodings and offering an open alternative to licensed datasets like CLLI, The Aleph demonstrates its value in scenarios requiring broader coverage and higher accuracy. The Aleph introduces a more adaptable method for handling complex cases, paving the way for future advancements in geolocation transparency and accuracy.

7 Discussion

In this section, we discuss some of the limitations of *The Aleph*'s LLM-approach and the data we rely on as input. First, DNS PTR records can often contain outdated or incorrect information, leading to errors in geographic inference [43]. In addition, a specific substring extracted from a PTR record could map to many locations, depending on the context (e.g., the hint 'mi' could encode the city Miami, Milan or the state Michigan). We expect outdated and incorrect information to be relatively uncommon. Validation experiments similar to those in Sec. 5.2 and Luckie et al. [25] may help increase confidence in the generated regular expressions and geolocation hints. Similarly, we expect LLMs to be able to disambiguate cases where a geolocation hint may point to multiple locations from the context they have through their training data [30].

Another potential issue is the impact of geographic information being embedded in languages other than English. For example, some operators may use non-English words for city names or other geographic markers in their PTR records. Recent work [39] shows that large language models,

such as GPT-3 and PaLM, exhibit strong reasoning abilities across multiple languages, even in underrepresented languages like Bengali and Swahili.

Finally, LLMs, including those used for extracting geographic hints, may "hallucinate" information, leading to errors in the extracted geolocations [2, 18]. Our prompting strategies and hyperparameters are designed to minimize the risk of hallucination by carefully structuring inputs and guiding model outputs. These strategies are informed by understanding that while intrinsic hallucinations result from content directly contradicting the input, extrinsic hallucinations involve generating additional, potentially plausible but unverified, information [2, 18].

8 Related Work

The challenges of geolocation in the Internet and efforts to leverage hints in PTR records associated with network devices as a long history, going at least as far back as early 2000s with GTrace [34], IP2geo [33], and Rocketfuel [40]. GTrace [34] leveraged geographical hints in node names, such as city names or airport codes, to build a graphical visualization of traceroute, IP2geo [33] extended this approach with the addition of host localization, and Rocketfuel leveraged it to map the router-level topology of the Internet, while extending it with manually generated regular expression to extract geohints [40].

Recent efforts have tried to automate the task of extracting PTR geolocation hints [13, 23, 25–27, 38]. HLOC (Hints-Based Geolocation Leveraging Multiple Measurement Frameworks) uses a prefix tree to match segments of DNS names against a detailed dictionary of geographic codes and active probing as part of its generation phase of the extracting rules. DRoP [13] tries to automate part of the task by searching for geographic encoding based on a previously populated dictionary of geographic-related strings.

More recently, Luckie et al. [25] automatically extract and interpret geo-hints embedded into hostnames using regexes informed by a dictionary that includes strings such as airport codes, city, state and country names), and learn simple deviations from geohints such as prefix and partial matches. Hoiho results from numerous efforts to extract various types of encoded information from DNS PTR records, such as ASNs, network names, and geolocation hints, and has become the state-of-the-art tool for extracting embedded geographic information from these records. Despite its high accuracy, Hoiho has relatively low coverage, due in part to its limitations in capturing unconventional geographic hint. Ovidiu et al.[10] finds marginally more complicated hints by training a binary classifier on a test set of locations, but it is limited by its inability to disambiguate hints that could point to multiple locations and focus on end-user ips.

Our work builds on this extensive line of research and the observation that LLMs, rather than humans, are better equipped to identify patterns in DNS PTR records and create extraction rules.

9 Conclusions and Future Work

Internet geolocation has long been a challenging problem, hindering research in various fields. We propose an LLM-based approach to extract geo-hints from DNS PTR records, reducing the reliance on manual efforts. Our analysis shows that 58% of operators encode geographic information in some of their PTR records, with formats varying within and between operators. We extracted geographic information and validated it using ground truth data from operators and active probing. We evaluate the effectiveness of our approach, applying the set of inferred regular expressions by *The Aleph*, and compare our results with those of Hoiho and GeoFeeds on a publicly available Internet topology dataset. We make *The Aleph* publicly queryable and invite the community to extend our hint mappings and regular expressions. Future work may include automating hint-mapping and class definition extensions by querying the LLM for unmapped hints and records that do not fit into any class, and building RTT based hint-validation pipelines to enhance accuracy.

10 Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable feedback in improving the quality of the paper. Special thanks to Jason Livingood (Comcast) and Romain Fontugne (IIJ) for their assistance with validation and to Raffaele Sommese (University of Twente) for his support with the OpenIntel database. This work was supported by National Science Foundation grants CNS-2107392 and CNS-2246475.

References

[1] 2024. World Bank Regions. https://www.worldbank.org/en/where-we-work
[2] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv* (2023).
[3] Timm Böttger, Felix Cuadrado, and Steve Uhlig. 2018. Looking for hypergiants in peeringDB. *ACM SIGCOMM CCR* (2018).
[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* (2020).
[5] CAIDA. 2019. Macroscopic Internet Topology Data Kit (ITDK). http://www.caida.org/data/internet-topology-data-kit/.
[6] CAIDA. 2025. Archipelago Measurement Infrastructure Updates. https://catalog.caida.org/details/media/2011_archipelago. Accessed: 2025-1-15.
[7] Massimo Candela. 2022. geofeed-finder. https://github.com/massimocandela/geofeed-finder.
[8] Esteban Carisimo, Carlos Selmo, J Ignacio Alvarez-Hamelin, and Amogh Dhamdhere. 2019. Studying the evolution of content providers in IPv4 and IPv6 internet cores. *Computer Communications* (2019).
[9] Joseph Chabarek and Paul Barford. 2013. What's in a name? decoding router interface names. In *Proceedings of HotPlanet*.
[10] Ovidiu Dan, Vaibhav Parikh, and Brian D. Davison. 2021. IP Geolocation through Reverse DNS. *ACM Trans. Internet Technol.* (2021).
[11] Rahel A. Fainchtein1 and Micah Sherr. 2024. You can Find me Here: A Study of the Early Adoption of Geofeeds. In *Proc. of PAM*.
[12] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv* (2023).
[13] Bradley Huffaker, Marina Fomenkov, and kc claffy. 2014. DRoP: DNS-based router positioning. *ACM SIGCOMM CCR* 44, 3 (jul 2014).
[14] Geoff Huston. 2014. How Big is that Network? https://labs.apnic.net/?p=526.
[15] IANA. [n. d.]. Address and Routing Parameter Area (arpa). https://www.iana.org/domains/arpa. Accessed: 2024-05-13.
[16] Mattia Iodice, Massimo Candela, and Giuseppe Di Battista. 2019. Periodic path changes in RIPE atlas. *IEEE Access* 7 (2019), 65518–65526.
[17] Costas Iordanou, Georgios Smaragdakis, Ingmar Poese, and Nikolas Lautaris. 2018. Tracing Cross Border Web Tracking. In *Proc. of IMC*.
[18] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* (2023).
[19] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. 2009. Nation-state routing: Censorship, wiretapping, and BGP. (2009).
[20] Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv* (2023).
[21] Youndo Lee and Neil Spring. 2017. Identifying and analyzing broadband internet reverse DNS names. In *Proc. of CoNEXT*.
[22] Dave Levin, Yundo Lee, Luke Valenta, Zhihao Li, Victoria Lai, Cristian Lumezanu, Neil Spring, and Bobby Bhattacharjee. 2015. Alibi Routing. In *Proc. of ACM SIGCOMM*.
[23] Matthew Luckie, Bradley Huffaker, and k claffy. 2019. Learning Regexes to Extract Router Names from Hostnames. In *Proc. of IMC*.
[24] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. 2013. AS relationships, customer cones, and validation. In *Proc. of IMC*.

[25] Matthew Luckie, Bradley Huffaker, Alexander Marder, Zachary Bischof, Marianne Fletcher, and K Claffy. 2021. Learning to extract geographic information from internet router hostnames. In *Proc. of CoNEXT*.
[26] Matthew Luckie, Alexander Marder, Marianne Fletcher, Bradley Huffaker, and K. Claffy. 2020. Learning to Extract and Use ASNs in Hostnames. In *Proc. of IMC*.
[27] Matthew Luckie, Alexander Marder, Bradley Huffaker, and k claffy. 2021. Learning Regexes to Extract Network Names from Hostnames. In *Proc. of IMC*.
[28] Matthew J. Luckie. 2024. Geolocation Data Analysis February 2024. https://users.caida.org/~mjl/rnc/geo/202402-geo/. Accessed: 2025-01-15.
[29] OpenAI. 2024. OpenAI Platform: GPT-4 Documentation. https://platform.openai.com/docs/models/gpt-4. Accessed: 2024-05-14.
[30] OpenAI and Josh Achiam et al. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
[31] OpenINTEL. 2024. OpenINTEL rDNS Dataset. https://www.openintel.nl/data-access/
[32] Ramakrishna Padmanabhan, Aaron Schulman, Dave Levin, and Neil Spring. 2019. Residential links under the weather. (2019).
[33] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. 2001. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proc. of ACM SIGCOMM*.
[34] Ram Periakaruppan and Evi Nemeth. 1999. GTrace - A Geographica traceroute tool. In *Proc. of USENIX Lisa*.
[35] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. 2011. IP Geolocation Databases: Unreliable? 41, 2 (April 2011).
[36] RIPE NCC. [n. d.]. Configuring Reverse DNS. https://apps.db.ripe.net/docs/Database-Support/Configuring-Reverse-DNS/. Accessed: 2024-05-13.
[37] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv* (2023).
[38] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. 2017. HLOC: Hints-based geolocation leveraging multiple measurement frameworks. In *Proc. of TMA*.
[39] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057* (2022).
[40] Neil Spring, Ratul Mahajan, and David Wetherall. 2002. Measuring ISP topologies with Rocketfuel. In *Proc. of ACM SIGCOMM*.
[41] Z. Weinberg, S. Cho, N. Christin, Vyas Sekar, and Phillipa Gill. 2018. How to Catch when Proxies Lie: Verifying the Physical Locations of Network Proxies with Active Geolocation. In *Proc. of IMC*.
[42] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. 2007. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts.. In *NSDI*.
[43] Ming Zhang, Yaoping Ruan, Vivek S Pai, and Jennifer Rexford. 2006. How DNS Misnaming Distorts Internet Topology Mapping.. In *USENIX Annual Technical Conference, General Track*. 369–374.

Appendix Organization The appendix is organized is follows. Section A provides instructions for using and extending *The Aleph*. Section B provides a detailed comparison of the performance of *The Aleph* and Geofeeds on `itdk2-2024`. Section C lists the top providers by the number of different classes of PTR records they have. Section D lists the top 20 cities by the number of geo-hints, providing examples for each city. Section E contains RTT distributions up to the 95th percentile for various providers across different regions. Each figure shows the empirical cumulative distribution function (CDF) of a provider in our validation set.

A How to Contribute

We applied *The Aleph* to a set of 2,646 ASes, collected a large database of regular expressions and hints, and validated a subset of them. We discuss these results in Sec. 5. We make this database publicly queryable through a website at https://thealeph.ai, which also hosts a RESTful api (https://thealeph.ai/docs) and details about *The Aleph*'s prompts and raw output for AT&T.

Beyond validation, we hope this will encourage community contributions to expand this *The Aleph* dataset. We will manually curate all contributions, to ensure the accuracy and quality of the data, before adding them to the repository.

B *The Aleph* and GeoFeeds

We compare the number of IP's geolocatable by *The Aleph* and Geofeeds. The Geofeed dataset is composed of a mapping of prefixes to locations. We check every IP address in the `itdk-2024` dataset, and count the number of IPs covered by prefixes in our GeoFeed dataset. We then provide all the IP addresses along with their ASN and PTR information as input to *The Aleph*, and count the number of IP addresses for which it extracted a location. To compare *The Aleph* and GeoFeeds, we focus on the 306 ASNs for which both methods provide information. Table 12 presents the results, both methods across the complete dataset and the two focused subsets.

Table 12. Comparison of metrics between *The Aleph* and GeoFeeds. We include results for the full dataset and two subsets for which both methods have information.

Tool	*The Aleph*	GeoFeeds
Complete View		
ASNs	2,646	3,358
PTR records (geolocation info.)	480,906	26,724 (1,129,911)
Focused View (overlap with *The Aleph*)		
ASNs	—	306
PTR records (geolocation info.)	—	15,382 (750,132)
The Aleph	—	562

The Aleph has regular expressions for 2,646 ASes, as described in Sec. 4, and extracts geographic information from 480,906 PTR records from the complete `itdk-2024` dataset, mapping to 1,806 unique cities worldwide. GeoFeeds, on the other hand, confirms locations for 1,129,911 IP addresses associated with 3,358 ASes, with only 26,724 (2%) having associated RDNS records. Focusing on the 306 overlapping ASes between *The Aleph* and GeoFeeds, GeoFeeds can locate 15,382 IP addresses with associated PTR records. From these, *The Aleph* extracts geographic hints from 562 records (3.6%) associated with 25 ASes, while Hoiho extracts information from none. The small percentage of IPs covered by GeoFeeds with associated PTR records suggests that the information gathered from these two methods are complementary, as GeoFeeds primarily covers prefix-based geolocation, whereas *The Aleph* focuses entirely on extracting geographic hints from PTR records.

C Top Providers By Number of Regular Expressions

Table 13. Provider Classifications and Examples

Provider (ASN)	Number of Classes	Example
Multnomah Education District (32522)	15	autohost66-154-154-222.seaside.k12.or.us
Orange - Côte d'Ivoire (29571)	12	lsci2m-154.68.47.133.aviso.ci
Gabon Telecom (16058)	10	speedtest.gabontelecom.ga
Sudatel (15706)	10	topografix.maps.sudani.sd
Telecom Algeria (36947)	10	mail.univ-tlemcen.dz
Cogent (174)	9	243-pool4.ras15.gaatl-i.alerondial.net
Verizon (701)	10	static-98-116-129-183.nycmny.fios.verizon.net
AT&T (7018)	7	adsl-99-163-197-39.dsl.wac2tx.sbcglobal.net
Amazon (16509)	7	ec2-34-212-135-253.us-west-2.compute.amazonaws.com
Google (15169)	6	svo04s30-in-f47.1e100.net

D Top-20 cities by number of geo-hints

Table 14. Top-20 cities by number of geo-hints

City	#	Examples
Tokyo	123	cloud, tokyo, nipr, tuat, icrr, to, ab, tk2, biochem, ga
Los Angeles	69	lvw-losangeles, uscalax, nup, norrissci, cal-voip-0223, tc8, .ucla.edu, lax08, eup, tc9
London	68	londen, uklon, londonah, ldncng, lon7, lon, tvu, bbk, uklond, LDN2
Chicago	98	crh, cen, uschi, rrh, lib, chic, chi1, ch1-agg-1, lpc, CHGO
Houston	84	bchs, humres, facct, webtech, parking, hstntx, biology, cmts8, EGR, hous
Paris	78	obspm, east-par, par03, u-paris, u-paris5, dc3-2, fr-par, parsfr, Paris1, u-paris2
New York	56	lga2, new, NewYork, yny1, cpmc, nynyc, nycmny, unyc, bobst, nym
Singapore	51	xsp2, sin6, sgp02, c02, sgp, SGP, sin100, wco1.sg, ifx, ih4-singapore
Dallas	39	dal, dald61, dald74, dfw04, dalb189, dall, dald11, dalb116, dfw6, Dallas
Frankfurt	35	fra, Frankfurt, fr4, fra8, f2c-frankfurt, uni-frankfurt, fra8-2, fr5, rf, fra4.de
San Jose	31	sjc-ca, sj, sj2, sjdc, san-jose, SJC, SanJose, sanjose, sjs, msj
Atlanta	29	gaatl, ATL, atln, atlm, Atlanta, coda, Atlanta1, atl10, lawn, mc.at
Seattle	22	washington, sttlwax1, pr0.sea20, sea1, SEA, sttl, sttn, sttlwa, chem, Seattle1
Miami	21	clipper, mia1.us, mi, usmia, m, mai, mim, grtmiana, miami.fl.us, Miami
Philadelphia	22	ist, ph, 267, Philadelphia, hap20, phlapa, phl1, Philadelphia1, phl, phla
Kuala Lumpur	15	kul, krt, klj03, psg, klj01, klj, eciti, wpj, kuala-lumpur, ebesi
Cairo	15	egyptian-steel, crystalegypt, cairo, edita, gest, elsharkawy, mwri, energyasteel, nbk, banquemisr
Ashgabat	13	cybersec, senagatbank, bashbina, ashgabat, constructionprice, tbbank, turkmenistanairlines, onko, tstb, minenergo

E Validation CDFs

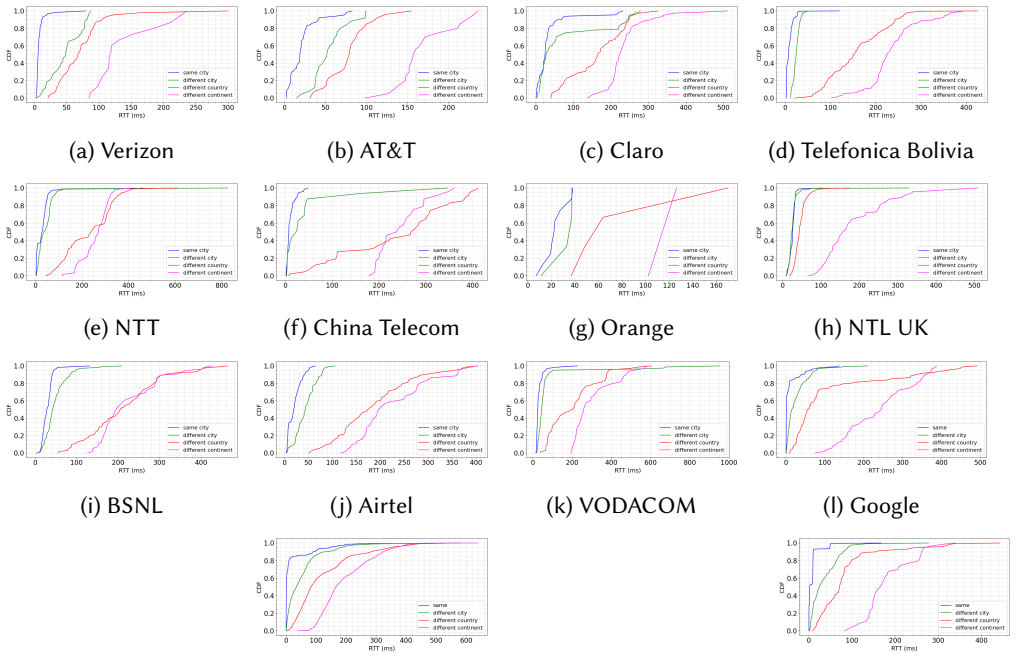


Fig. 8. Consolidated CDF distributions for various providers.