UNIKASSEL
VERSITÄT

Universität Kassel

Distributed Systems Group

Bachelor Thesis

Geolocation and map-based visualization of autonomous systems

Kai Liebscher

Mat.-Nr.: 31217260
liebscher.kai@gmail.com

supervised by
Bernd Spiess
Prof. Dr. rer. nat. Oliver Hohlfeld
Dr. Jens Kosiol

2024-02-01

Abstract

The Internet is global. It consists of tens of thousands of autonomous systems (ASes), some of which are also operating across the entire planet, while others are only present in a single country.

Knowing the regions in which ASes are operating is beneficial for players in the internet industry. However, determining the geolocation of an AS is not trivial. While it is possible to look up in which country an ASes is registered, this does not necessarily tell where it is actually operating.

In this work, geographic location information from an Internet Protocol (IP) geolocation database is combined with IP prefixes announced by ASes. The goal is to achieve a more sophisticated answer to the question: Where is an AS operating?

Additionally, a web application is created which visualizes the results on an interactive map and in several dynamic charts.

Contents

1 Introduction

The Internet consists of autonomous systems (ASes). They connect to each other in order to form a global network. Not all ASes are equal, though. They differ in size and the area they are operating in. Data can be sent over the Internet from one end of the world to the other and needs to pass several ASes for that.

Since this requires AS resources, AS operators can decide whether or not to allow data passing through their network. This decision often involves money, usually when two neighboring ASes differ in size. The smaller, more regional, tier-2 or tier-3 AS will pay the larger AS with greater international reach in exchange for traffic routing through their network. This is called Internet Protocol (IP) transit [1, p. 41].

Alternatively, ASes can decide to connect directly without an intermediate, higher tier AS. This is a service that is offered by Internet exchange points (IXPs).

1.1 Geolocating autonomous systems is valuable

IXPs need to make business decisions based on how many ASes are operating in a country. Since each local ASes is a potential customer of the IXP, the number of ASes operating in a country is proportional to the market potential. From this perspective, knowing how many ASes are operating in a country becomes valuable information.

1.2 Geolocating autonomous systems is ambiguous

ASes are registered within a country. Looking up the country or registration gives a hint where an AS might be operating. However, this hint might be only partially true or completely false.

A less trivial, but potentially more accurate approach involves locating the IP prefixes an AS announces by matching the contained IP addresses with an IP geolocation database.

1.3 Research question

This work answers the questions: Can we utilize IP geolocation databases to get a clearer view of where ASes are operating? Given a country, how many ASes are operating there?

2 Background

Where are autonomous systems (ASes) operating? In order to find a sophisticated answer to this question, a number of data sources will be used and combined.

Comprehending the approach of this paper requires to know about ASes and the role they play in the composition of the Internet. A basic knowledge of how the Border Gateway Protocol enables routing between ASes is needed. It is also necessary to understand Internet Protocol (IP) addresses, IP prefixes and IP geolocation databases.

2.1 Autonomous Systems

ASes are the building parts of the Internet. They are independent networks administered by companies. Depending on their size and area they are operating in, they connect to each other as transit partners or as peering partners.

The Internet Assigned Numbers Authority (IANA) is responsible for managing ASes. Their activities include assigning official autonomous system numbers (ASNs) to ASes and granting them blocks of IP addresses for addressing devices within their network. These responsibilities are usually delegated to five regional Internet registries (RIRs). The RIR responsible for Europe is the Réseaux IP Européens Network Coordination Centre – European IP Networks Network Coordination Centre (RIPE NCC).

2.2 IP prefixes and BGP routing

RIRs assign blocks of IP addresses to owners of ASes. These blocks are bundled in a shorthand notation called IP prefixes. The owner may assign those addresses to devices in their network so that IP data packets can be addressed to them.

In order for this to work globally, packets from outside the destination AS also need to know how to reach the AS. This is solved by routers of different ASes communicating to each other via the Border Gateway Protocol (BGP). Using BGP, ASes can announce IP prefixes at which their clients are reachable. BGP routers in other ASes remember and propagate that information so that they can build an exhaustive routing table containing AS routes to any announced IP prefix.

2.3 IP geolocation databases

IP geolocation databases link IP addresses to geographic locations. The providers of such databases obtain their information through triangulation using latency measurements among other methods. This leads to a varying accuracy of the locations that the databases offer. Some IP addresses are located accurately enough to be seen in a specific city with certainty while others can at least be associated with a country [2].

IP geolocation databases have various usages. Content providers may use the location of a user – that is, the location of their IP address – in order to restrict access to their media to certain regions. Personalized advertising is another common use case. In this work, an IP geolocation database is used to attach geographic locations to BGP IP prefix announcements.

3 Related Work

In existing tools, the registration countries of autonomous systems (ASes) can be looked up.

- Hurricane Electric[1] lists them as *Country of Origin*.
- Cloudflare[2] also lists countries of registration.
- The Réseaux IP Européens Network Coordination Centre – European IP Networks Network Coordination Centre (RIPE NCC) offers a complete list[3] of ASes with their registration countries in a machine readable format.

This information can be used to determine how many ASes are operating in a given country. In this work, it is assumed and shown that this method is insufficient and misses a significant amount of ASes that are also operating in the given country, but registered elsewhere.

[1] `https://bgp.he.net`
[2] `https://radar.cloudflare.com`
[3] `https://ftp.ripe.net/pub/stats/ripencc/nro-stats/latest/nro-delegated-stats`

4 Approach

Where is an autonomous system (AS) operating? This question can be answered in different ways. The simplest way is by looking up in which country an AS is registered. However, the fact that an AS is registered in a given country does not guarantee that its infrastructure is operated in the same country. In fact, it is also possible that an AS operates in several countries at the same time. It might even operate globally, being present in virtually every country in the world. This shows that it is not sufficient to only take into account where an AS is registered. Utilizing additional data sources to get a clearer picture is the goal of this approach.

One such additional data source could be found in considering the Internet exchange points (IXPs) at which an AS is peering. This might introduce more relevant countries to a single AS since it might be peering at IXPs in several countries. However, AS operators are not obliged to disclose that information. So, while this method could yield more thorough results than looking up the country of registration, it could also just fail when an AS operator decides to keep their peering activity information private.

A similar approach involves determining the geographic locations of the infrastructure of an AS. This requires knowledge about the Internet Protocol (IP) addresses used by an AS. It can be obtained by inspecting the IP prefix announcements of individual ASes as collected by Border Gateway Protocol (BGP) routers. In order for routing between ASes to be possible, BGP routing information must be public. Combined with an IP geolocatin database, every IP prefix and thus every AS can be linked to geographic locations. In this work, this promising approach is implemented and evaluated with the goal of creating a comprehensive view of the geographical situations of ASes.

4.1 Data sources

As outlined in the previous section, several data sources are needed in order to implement the approach. This section lists and describes the data sources that are going to be used in the implementation.

Table 4.1: Used data sources

Data source	Relevant information
① AS list from RIPE NCC	autonomous system number (ASN) + country of registration
② BGP route collector dump	ASN + announced IP prefixes
③ MaxMind GeoIP	IP addresses + geographic locations
④ PeeringDB dump	ASN + AS name + IXPs

The Réseaux IP Européens Network Coordination Centre – European IP Networks Network Coordination Centre (RIPE NCC) offers a ① global list linking ASes to their country of registration.[1] This gives a first hint at where each AS might be operating. In some cases, where an AS is operating matches the country where it is registered. For those cases, looking up an AS of interest in this list would be sufficient.

In other cases, an AS might be operating in additional countries other than where it is registered. It might not even be operating in its country of registration at all. This can be examined by consulting additional data sources.

[1] `https://ftp.ripe.net/pub/stats/ripencc/nro-stats/latest/nro-delegated-stats`

Also available from RIPE NCC are ② BGP route collector dumps.[2] The entries of such dumps contain various routing information such as IP prefix announcements and the AS they originate from.

Another useful data source is ③ MaxMind's GeoIP database.[3] It is an IP geolocation database that links IP addresses to geographic locations. An account with the company is necessary in order to download the database.

We also process information from a ④ PeeringDB dump.[4] This dataset contains names for ASes and links them to IXPs they are known to be peering at.

4.2 Combining data

Each of the used data sources contributes a puzzle piece to the dataset that is defined as the goal of the approach of this work.

Joining the entries in ① RIPE NCC's AS list with the entries in the ② BGP route collector dump via their respective ASN fields yields a list of ASes with their registration country and a list of announced IP prefixes.

Joining the result with ③ MaxMind's GeoIP database extends the result with a list of geographic locations that are associated with each AS. Furthermore, the corresponding country is added to each of those locations.

A list of countries where an AS is operating can be derived from the resulting dataset directly. Also, the dataset can be converted to a list mapping countries to lists of ASes operating there, thus answering the research questions posed in section 1.3.

For convenience, a ④ dump of the PeeringDB is utilized as source for names of ASes. It also lists IXPs at which an AS is peering in some cases. While that information could be used to further complete the list of countries where an AS is active in case PeeringDB indicated peering activity in a country where MaxMind did not detect any of the announced prefixes, this is subject to future work (section 7.3).

Everything related to downloading, extracting and merging data will be covered by the term *backend*. Its implementation is explained in section 5.1.

4.3 Visualizing the results

After acquiring and merging the various data sources into one large dataset, the result is difficult to make sense of. This is why a visualization of the resulting data is crucial to make it more understandable and evaluable.

Since this work is about geographic locations, a map is an obvious visualization tool. An interactive component enables a user to search all processed ASes and select one to see all associated locations on the map. Additionally, various charts will allow statistical analysis and evaluation.

Everything related to user interaction and visualization will be covered by the term *frontend*. Its implementation is explained in section 5.2.

[2] `https://data.ris.ripe.net/rrc00/latest-bview.gz`

[3] `https://download.maxmind.com/app/geoip_download?edition_id=GeoLite2-Citylicense_-key=<KEY>suffix=tar.gz`

[4] `https://publicdata.caida.org/datasets/peeringdb/year/month/peeringdb_2_dump_year_month_-day.json`

5 Implementation

The implementation consists of two main parts: Data is fetched, extracted, aggregated and optionally served in the backend. The results can then be displayed on an interactive map and in dynamic charts in the frontend.

5.1 Backend

Acquiring and combining the data takes place in the backend in a number of steps which involve downloading datasets from various external sources, extracting them and merging them. These steps are executed by several scripts that eventually generate the static data that can then be consumed by the frontend.

The scripts were written in Python to ensure modularity and extendability. The intermediate and final datasets use the JavaScript Object Notation (JSON) format since it is natively supported by Python (using a built-in library) and the frontend, being written in JavaScript, can also process this format directly.

There is a helper bash script `all/download_and_prepare_data.sh` that executes these scripts in the required order. This script can also be seen as a table of contents for the backend.

Listing 5.1: `all/download_and_prepare_data.sh`

```
# Download data from PeeringDB, MaxMind and RIPE
../download/download_bgpdump_bview.py "$@"           # section 5.1.1.1
../download/download_maxmind_geolite2_city.py "$@"   # section 5.1.1.2
../download/download_rir_statistics.py "$@"          # section 5.1.1.3
../download/download_peeringdb_2_dump.py "$@"        # section 5.1.1.4

# Extract relevant data
../extract/extract_bgpdump.py "$@"                   # section 5.1.1.1
../extract/extract_maxmind_geolite2_city.py "$@"     # section 5.1.1.2
../extract/extract_peeringdb_facilities.py "$@"      # section 5.1.1.4
../extract/extract_peeringdb_nets.py "$@"            # section 5.1.1.4
../extract/extract_peeringdb_nets_facilities.py "$@" # section 5.1.1.4
../extract/extract_peeringdb_organizations.py "$@"   # section 5.1.1.4

# Merge data
../merge/merge_nets_rir_statistics.py "$@"           # section 5.1.2.1
../merge/merge_nets_bgpdump_prefixes.py "$@"         # section 5.1.2.2
../merge/merge_nets_facilities.py "$@"               # section 5.1.2.3
../merge/merge_nets_maxmind.py "$@"                  # section 5.1.2.4
```

Each of those scripts is described in more detail in the following sections.

5.1.1 Fetching and extracting data

Several scripts for fetching data reside in the `download` directory. They utilize a utility script `download/util/download_file.py` that enables file downloads with a caching function.

Some downloaded files have to be extracted or contain unnecessary data that should be stripped before they can be processed further. Corresponding scripts are found in the directory `extract`.

5.1.1.1 Fetching and extracting a BGP dump

Border Gateway Protocol (BGP) route collector dumps are available from the Réseaux IP Européens Network Coordination Centre – European IP Networks Network Coordination Centre (RIPE NCC). They contain Internet Protocol (IP) prefix announcements that can be linked to an originating autonomous system (AS) by examining their AS path.

The download script `download/download_bgpdump_bview.py` downloads a dump file in Multi-Threaded Routing Toolkit (MRT) format. The extract script `extract/extract_bgpdump.py` then uses a third-party library `mrtparse` in order to parse the binary contents of the file as BGP messages.

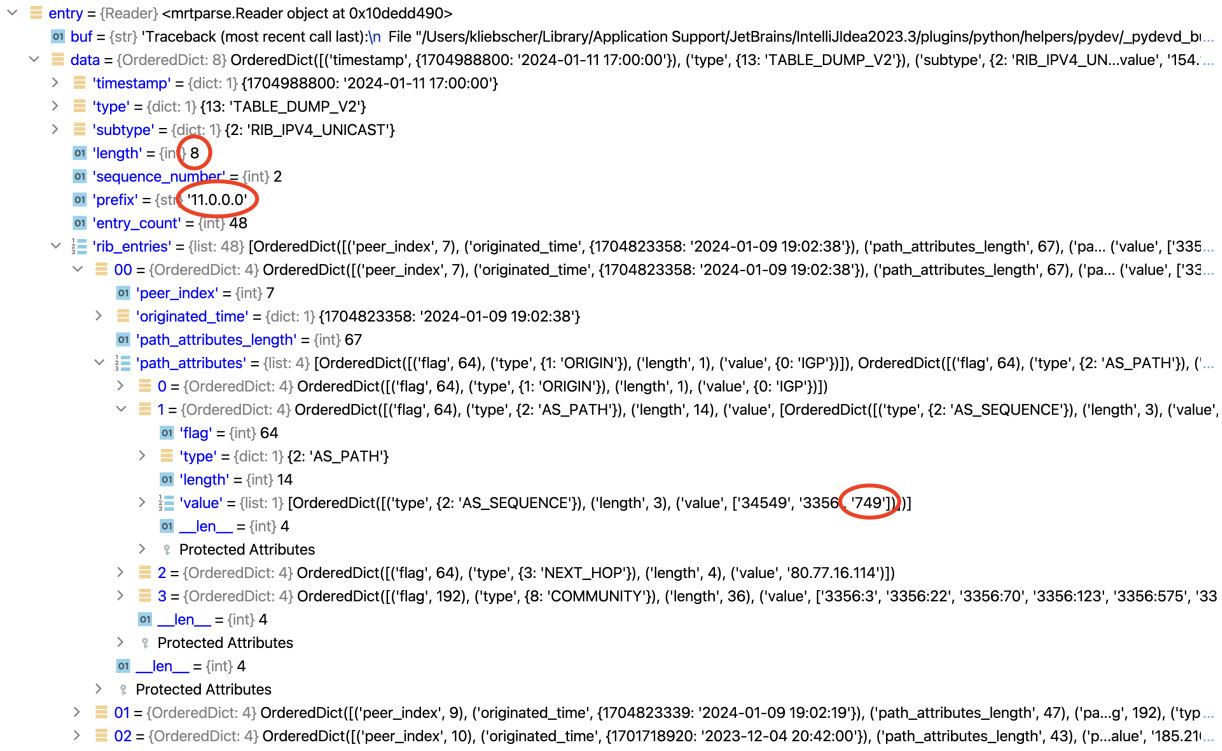


Figure 5.1: Example `RIB_IPV4_UNICAST` message containing an IP prefix announcement

The example BGP message shown in figure 5.1 contains an IP prefix announcement for the IP prefix `11.0.0.0/8`. The path attribute `AS_PATH` resembles the various ASes this announcement has passed through before being captured by the route collector. The last entry in that path, 749, is considered as the origin AS of the announcement and thus the AS that accommodates the devices using addresses from the announced IP prefix.

For each `RIB_IPV4_UNICAST` message, the announced IP prefix and the origin AS is recorded. The script then writes a JSON object mapping autonomous system numbers (ASNs) to an array of prefixes that the AS announced to the file `extracted/bgpdump.json`.

Listing 5.2: Example `extracted/bgpdump.json` generated by `extract/extract_bgpdump.py`

```
{
  "749": [
    "7.0.0.0/8",
    "11.0.0.0/8",
    ...
  ],
  ...
}
```

This list of IP prefix announcements is processed further at the merging stage as described in section 5.1.2.2.

5.1.1.2 Fetching and extracting MaxMind GeoIP

MaxMind's GeoIP database contains IP addresses and corresponding geographic locations. The database can be downloaded using a license key which can be obtained by signing up with MaxMind free of charge.

The download script `download/download_maxmind_geolite2_city.py` downloads the archived database in proprietary mmdb format. It is then extracted by the extraction script `extract/extract_maxmind_geolite2_city.py`. Afterwards, the database is ready to be used at the merging stage as described in section 5.1.2.4.

5.1.1.3 Fetching and extracting RIR registration information

The RIPE NCC lists all registered ASes and the country in which they are registered. This information will be used for comparison with the list of countries that are returned when looking up the announced IP prefixes in MaxMind's GeoIP database.

The download script `download/download_rir_statistics.py` downloads the list in the comma-separated values (CSV) format. At the merging stage, a script merges the registration country information as described in section 5.1.2.1.

5.1.1.4 Fetching and extracting data from a PeeringDB dump

PeeringDB links ASes to peering facilities and organizations. The backend uses it as source for the names of ASes. Its associated peering facilities will be displayed on the map.

The data from PeeringDB is available in the JSON format. The script `download/download_peeringdb_2_dump.py` downloads the latest dump. Several extract scripts then extract information about ASes and peering facilities from it.

With `extract/extract_peeringdb_facilities.py`, a global list of peering facilities is extracted from the dump. `extract/extract_peeringdb_nets.py` extracts all ASes each into a separate file. These AS-specific files will be extended with detailed information in the merging stage. When receiving a request for information concerning a specific AS from the frontend, these are the files that are going to be returned.

So far, no association between ASes and peering facilities was made. As this is a many-to-many relation, the PeeringDB dump has a separate list for that. The script `extract/extract_peeringdb_nets_facilities.py` extracts the associations between ASes and peering facilities into a separate file that is later used to extend each AS file with a list of peering facilities. This process is described in section 5.1.2.3.

Finally, `extract/extract_peeringdb_organizations.py` extracts all peering organizations. This information is currently not used.

5.1.2 Merging data

At the merging stage, data from the various sources are brought together. ASes are enriched with information about their registration country (section 5.1.2.1), facilities they are peering with (section 5.1.2.3), the IP prefixes they announce (section 5.1.2.2) and where MaxMind locates the IP addresses in those prefixes (section 5.1.2.4).

5.1.2.1 Merging registration country information

In this part, each AS is extended with a field indicating its registration country. For that, `merge/merge_nets_rir_statistics.py` iterates over the CSV file from RIPE NCC and reads

each line, parsing the ASN and its registration country.

Listing 5.3: Example `nro-delegated-stats` downloaded by `download/download_rir_statistics.py`

```
...
arin|US|asn|1|1|20010920|assigned|e5e3b9c13678dfc483fb1f819d70883c|e-stats
arin|US|asn|2|1|19910110|assigned|279fb28df10add3bd7028865951995a6|e-stats
...
ripencc|DE|asn|3320|1|19950223|assigned|bc56e635-34e3-4511-bfb3-2718085daf18|e-stats
...
```

Then, the corresponding autonomous system's file is loaded and extended by the new information. In case the file does not exist because it was not present in the PeeringDB dump, a new file is created.

5.1.2.2 Merging IP prefix announcements

The IP prefixes extracted from the BGP route collector dump earlier are now merged into the AS files. The script `merge_nets_bgpdump_prefixes.py` iterates over the intermediate file `bgpdump.json` (section 5.1.1.1) and adds each list of prefixes to the corresponding AS file. As a result, now for every AS there is a file containing its ASN, its country of registration (section 5.1.2.1) and a list of the IP prefixes it announces.

5.1.2.3 Merging peering facilities

Each AS is extended by a list of connected peering facilities that PeeringDB provides. The data has been extracted in section5.1.1.4 and is now merged into the individual AS files by the script `merge/merge_nets_facilities.py`.

5.1.2.4 Merging geolocation information from MaxMind GeoIP

At this point, each individual AS has a corresponding file containing a JSON object describing its ASN, its registration country, a list of IP prefixes announced by it and a list of peering facilities. The only piece of information missing in order to enable a sophisticated way of determining the countries each AS is operating in is the geographic locations of the IP addresses contained in the announced prefixes.

In order to obtain this data, the previously downloaded and extracted MaxMind GeoIP database is now queried for those locations. GeoIP comes with a Python library `geoip2.database` which allows direct queries to the database in the proprietary mmdb format.

The script `merge/merge_nets_maxmind.py` iterates over all individual AS files and then over each of its announced IP prefixes, querying GeoIP for all contained IP addresses.

GeoIP is queried with a single IP address. The geographic location returned by such a query varies in granularity regarding the amount of covered IP addresses. Some entries assign a location to just the requested IP address while in other cases one location stands for hundreds of IP addresses collectively (including the requested one). This fact serves as an opportunity for optimization. Querying every single IP address in an announced IP prefix would work, but take a lot of time, given the total amount of existing IP addresses. However, if a query result states that the returned location is part of a larger GeoIP entry covering 256 IP addresses, it would be meaningless to send additional queries for the remaining 255.

`merge/merge_nets_maxmind.py` has a implements a querying algorithm that makes use of this optimization. An iterator variable `current_address_to_look_up` is incremented by the number of IP addresses covered by the result after each query. If GeoIP does not return a result, an `AddressNotFoundError` is raised, `addresses_in_current_network` is set to 1 in the

`except` block and the iterator is also incremented by 1. Otherwise, the IP prefix of the result is considered to determine the value of `addresses_in_current_network`.

Since the result prefix can be even larger than the requested prefix, measures are taken to limit the amount of addresses taken into account. The last address of the prefix is obtained (`city.traits.network.broadcast_address`) but limited to the last address of the prefix being looked up. This way, addresses in the result prefix exceeding the requested prefix are not counted.

Listing 5.4: `merge/merge_nets_maxmind.py` **omits querying IP addresses if their location is already known**

```
def geolocate_prefix_in_maxmind(prefix_network):
  current_address_to_look_up = prefix_network.network_address
  last_address_to_look_up = prefix_network.broadcast_address

  while current_address_to_look_up < last_address_to_look_up:
    try:
      mm_city = city_reader.city(current_address_to_look_up) # might throw
          ↪ AddressNotFoundError
      mm_network = mm_city.traits.network

      last_address_in_mm_network = min(
        int(mm_network.broadcast_address),
        int(prefix_network.broadcast_address)
      )
      addresses_in_current_network = last_address_in_mm_network + 1-
          ↪ int(current_address_to_look_up)

    except AddressNotFoundError:
      addresses_in_current_network = 1

    current_address_to_look_up += addresses_in_current_network
```

All locations found for the announced IP prefixes are grouped and added to the AS file.

After the lookups have been finished, an overview of all ASes with summarized information is generated and saved to a single file. For each AS, it contains the ASN, the country in which it is registered, the number of IP addresses, the number of IP addresses located in the country of registration and the country in which most of the IP addresses were located. Additionally, an overview of all countries with condensed information and a separate file containing more detailed information for each country are generated. The AS and country overviews are transmitted to the frontend and used as a search index in order to enable instant search results after each keystroke without sending and having to wait for a network request. This is also described in section 5.2.2.

5.1.3 The final resulting dataset

After adding all geographic locations obtained from GeoIP to the dataset and generating overviews for countries and ASes, the resulting dataset is now ready to be consumed by the frontend.

For every AS, there is a separate file in `servable/net/<ASN>.json`.

Listing 5.5: Final 680.json

```
{
  "asn": 680,
  "name": "DFN Deutsches Forschungsnetz e.V.",
  "num_of_addresses": 7779582,
  "registered_country": "DE",
  "num_of_addresses_in_registered_country": 7777024,
```

```
  "mm_top_country": "DE",
  "prefixes": [
    "45.12.192.0/22",
    "84.246.64.0/21",
    ...
  ],
  "num_of_countries": 2,
  "ips_per_country": {
    "DE": 7777024,
    "null": 2302,
    "US": 256
  },
  "locations": [
    {
      "lat": 52.4186,
      "lon": 13.2696,
      "accuracy_radius": 200,
      "num_of_addresses": 8192
    },
    ...
  ],
  "num_of_facilities": 3,
  "facilities": [
    {
      "id": 58,
      "name": "Digital Realty Frankfurt FRA1-16",
      "country": "DE",
      "latitude": 50.11967,
      "longitude": 8.73463,
      "org_id": 8592
    },
    ...
  ]
}
```

Additionally, there is an overview for all ASes in `servable/nets.json`. This file serves as a search index for the frontend as described in section 5.2.2.

Listing 5.6: nets.json contains an overview for all ASes

```
{
  "2906": {
    "asn": 2906,
    "name": "Netflix",
    "registered_country": "US",
    "mm_top_country": "US"
    "num_of_countries": 30,
    "num_of_addresses": 160768,
    "num_of_addresses_in_registered_country": 122880,
    "num_of_facilities": 77,
  },
  "31337": {
    "asn": 31337,
    "name": "Eleet Networks",
    "registered_country": "SE",
    "mm_top_country": "SE"
    "num_of_countries": 1,
    "num_of_addresses": 1024,
```

```
    "num_of_addresses_in_registered_country": 1024,
    "num_of_facilities": 0,
  },
  ...
}
```

Another view on the data is also generated. With regard to the second research question (*Given a country, which ASes are operating there?*), for every country there is a file listing the present ASes. The frontend derives the AS table from these files.

Listing 5.7: ET.json contains an overview for all ASes and peering facilities in Ethiopia

```
{
  "country_iso": "ET",
  "country_name": "Ethiopia",
  "nets": [
    13335,
    15169,
    ...
  ],
  "facilities": [
    {
      "id": 12504,
      "name": "RAXIO Data Centre - Addis Ababa",
      "country": "ET",
      "latitude": 8.997753,
      "longitude": 38.779203,
      "org_id": 32779
    },
    ...
  ]
}
```

At last, there is a condensed overview for countries in `servable/countries.json`.

Listing 5.8: countries.json contains an overview for all peering facilities and ASes in Ethiopia

```
{
  "AR": {
    "country_iso": "AR",
    "country_name": "Argentina",
    "number_of_nets": 1172
  },
  "FJ": {
    "country_iso": "FJ",
    "country_name": "Fiji",
    "number_of_nets": 33
  },
  ...
}
```

5.2 Frontend

At this point, the data has been obtained and merged into a unified dataset. For every AS, there is a list of countries where its announced prefixes have been located by MaxMind GeoIP. This would already be sufficient in order to answer the research questions (section 1.3). However,

evaluating the generated dataset is difficult. In order to allow humans to grasp what the resulting dataset actually contains, different kinds of visualization have been implemented.

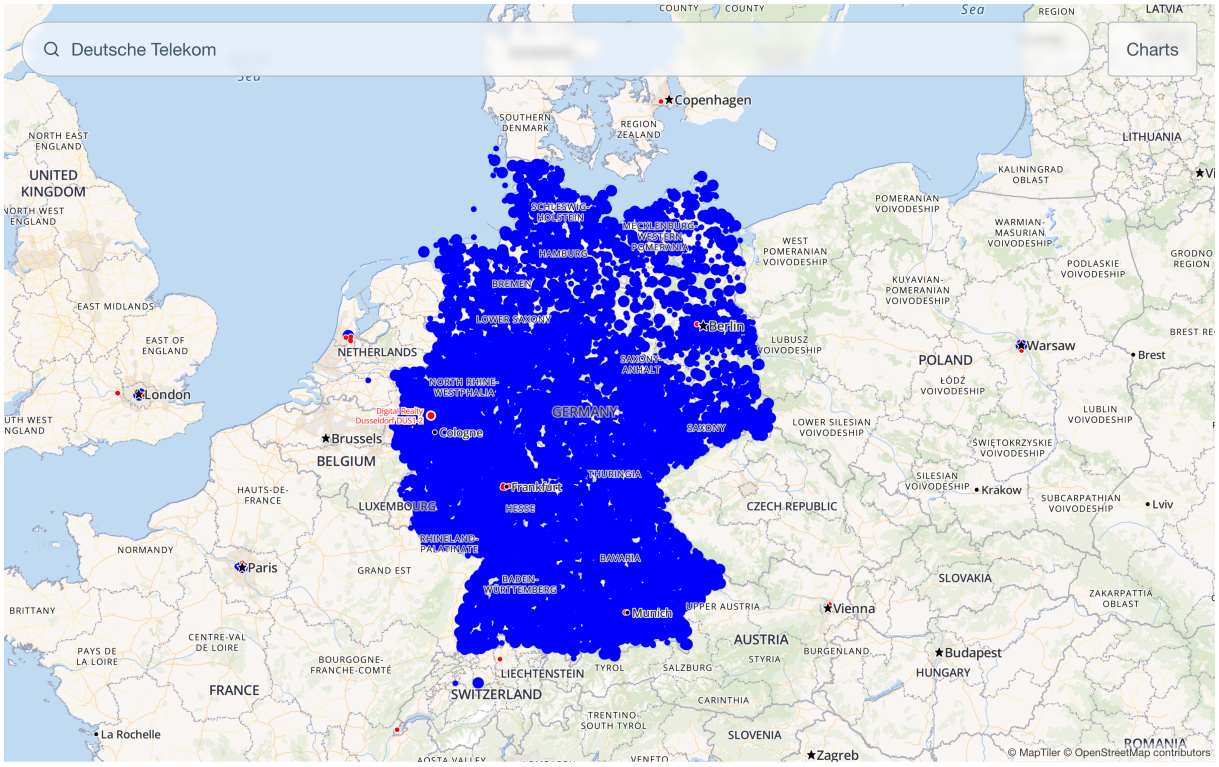


Figure 5.2: Frontend showing AS3320 (Deutsche Telekom) on a map

Figure 5.2 shows the implemented frontend in action after searching and selecting AS3320 (Deutsche Telekom AG). It consists of several components that are described in the following subsections.

A web application was chosen since an interactive map can be added effortlessly as described in section 5.2.1. Additionally, React was chosen as a widely supported framework to aid implementing interactive components and state management.

5.2.1 Map

This work revolves around geographic data. The obvious choice for visualizing geographic data is a map. The *OpenStreetMap* project is a geodatabase that comes with a large ecosystem of tools and libraries that enable developing map-based applications and other software related to geodata. This includes *maplibre-gl-js*, a JavaScript library suitable for embedding an interactive, draggable and zoomable map based on vector tiles into web applications. The map can be customized with individual styles and additional data layers.

In order to make the dataset generated in the backend visible, a React application with *maplibre-gl-js* was set up. An additional library *react-map-gl* wraps *maplibre-gl-js* in a React component, allowing to synchronize its view state (map position and zoom level) with a React state variable and adding map sources and layers declaratively. The map is embedded in the application's main component `<Map>`, defined in the file `Map.js`.

The geographic locations found in MaxMind GeoIP and the peering facilities found in PeeringDB are added to the map as layers. Maplibre layers are based on sources in GeoJSON format which are derived from the data that the backend sends. This is implemented in the files `LocationLayer.js` and `IconLayer.js`.

Only the locations associated with a selected AS are displayed on the map. Blue circles represent locations found by MaxMind as positions for IP prefixes. Their size correlates with the

number of IP addresses that were found at the location. Red dots represent peering facilities.

The map and its data layers help the human eye grasp the meaning of the generated data immediately. For example, the screenshot in figure 5.2 tells the user at first sight that the vast majority of the IP prefixes announced by AS3320 (Deutsche Telekom AG) are actually operating in Germany. Also, it becomes evident that there is some activity in neighboring countries, however on a smaller scale.

Since the map is only supposed to show locations associated with a chosen AS, the user needs to be able to choose. This requires an interactive component – a search bar.

5.2.2 Search

The search bar is the main entry point to the application, prominently placed at the top of the page. It is implemented in the file `Search.js`.

The user can enter an ASN, the name of an AS or a country – by its name or its ISO 3166-1 alpha-2 code – and is immediately presented with suggestions matching the entered search term as shown in figure 5.3.

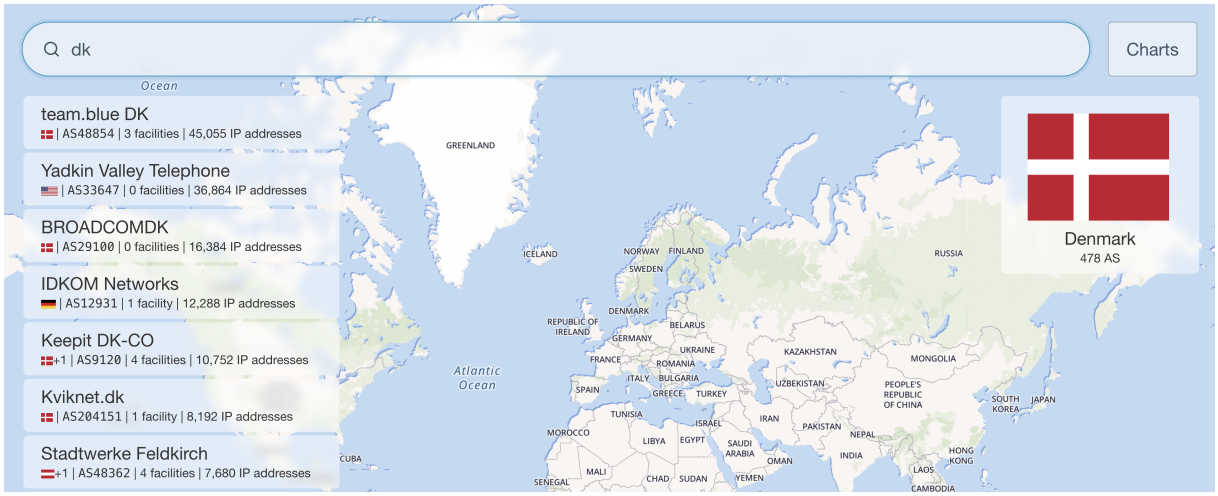


Figure 5.3: The search bar at the top makes instant suggestions

This is where the overviews generated in section 5.1.2.4 are used. Since the frontend loads the overviews from the server immediately, any search can be carried out locally in the frontend. There is no need to send requests to a server and wait for an answer which contributes to the user experience.

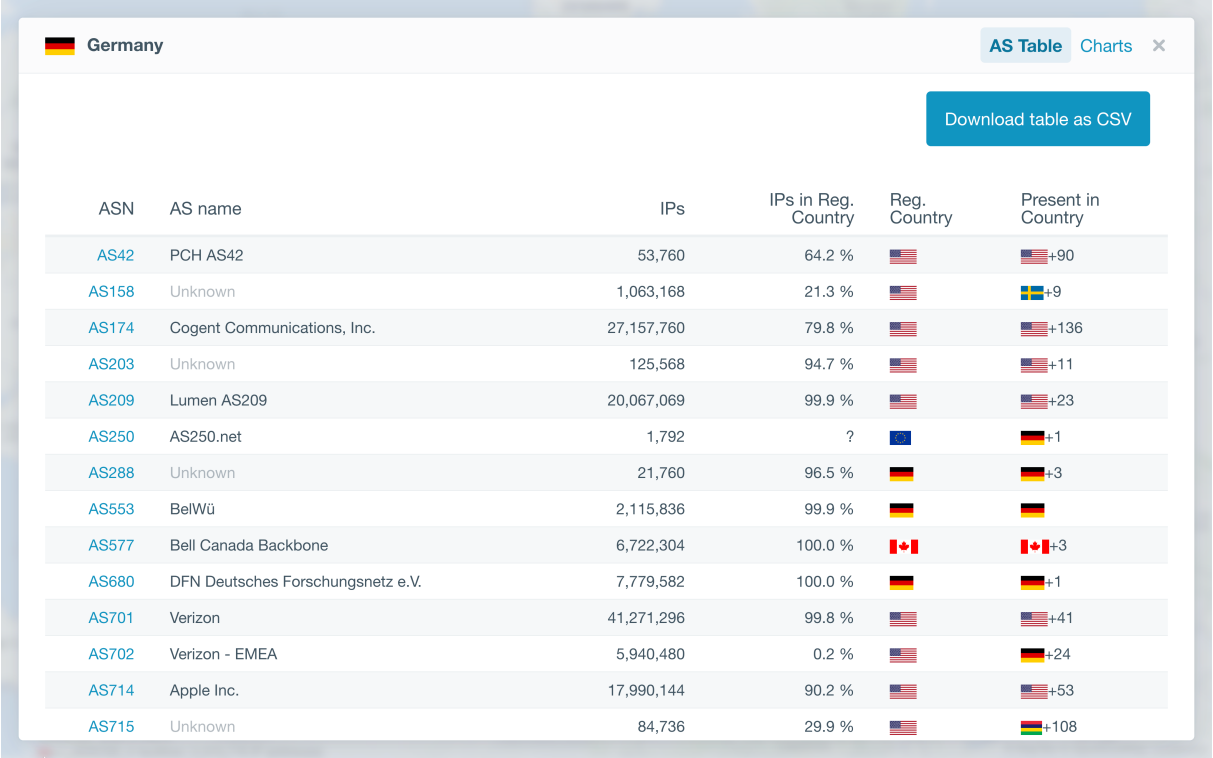
After typing into the search bar, the user is presented with matching ASes on the left side. Each AS is listed with its ASN, its name, the country where most of its announced IP prefixes have been located, the number of additional countries (if any) and the number of facilities and IP addresses. When choosing one of those suggestions, the suggestions disappear. The data layers for geolocated IP prefixes and peering facilities are shown and the map automatically zooms and pans to a view state so that the data layers are fully visible.

On the right side, the user also gets to choose from country suggestions. When choosing a country, a new dialog appears. It offers a table containing all ASes detected in the selected country (section 5.2.3) and dynamic charts (section 5.2.4).

5.2.3 Country AS Table

After selecting a country, the user is presented with a table (figure 5.4) containing all ASes that have been located in the selected country. This component is implemented in the file `CountryNetTable.js`.

Shown details include the ASN, the AS name, the number of IP addresses, the registration country, the share of IP addresses located in the registration country, the country in which most of the IP addresses have been located and the number of countries in which any number of IP addresses announced by the AS have been located.



ASN	AS name	IPs	IPs in Reg. Country	Reg. Country	Present in Country
AS42	PCH AS42	53,760	64.2 %		+90
AS158	Unknown	1,063,168	21.3 %		+9
AS174	Cogent Communications, Inc.	27,157,760	79.8 %		+136
AS203	Unknown	125,568	94.7 %		+11
AS209	Lumen AS209	20,067,069	99.9 %		+23
AS250	AS250.net	1,792	?		+1
AS288	Unknown	21,760	96.5 %		+3
AS553	BelWü	2,115,836	99.9 %		
AS577	Bell Canada Backbone	6,722,304	100.0 %		+3
AS680	DFN Deutsches Forschungsnetz e.V.	7,779,582	100.0 %		+1
AS701	Verizon	41,271,296	99.8 %		+41
AS702	Verizon - EMEA	5,940,480	0.2 %		+24
AS714	Apple Inc.	17,990,144	90.2 %		+53
AS715	Unknown	84,736	29.9 %		+108

Figure 5.4: Every country comes with a detailed table containg all detected ASes

This data can be interpreted quickly. Percentages in the column *IPs in Reg. Country* show how many of the addresses contained in the announced IP prefixes are located in the same country where the AS is registered. It becomes apparent that while for some ASes, the approach involving an IP geolocation database does not provide any new insights, in other cases almost every associated IP address is located outside of the registration country. For example, (almost) all the IP addresses in the prefixes announced by AS553 (BelWü) were located in Germany and nowhere else, so you can conclude that the AS is only operating in Germany.

On the other hand, AS702 (Verizon - EMEA) has only 0.02 % of its IP addresses located in the United States (where it is registered) while the remaining 99.8 % are being used in 24 other countries, of which Germany ranks first as displayed in the column titled *Present in Country*. This means that the AS has a significant presence in Germany and should be taken into account when asking for all ASes that are active in Germany. Other methods of putting together lists of ASes active in Germany would have missed this – and numerous others, as becomes even clearer in section 5.2.4.

If the user is interested in a specific AS in this list, they can click the ASN and are presented with the same map view they would have received if they had searched for the AS directly.

For convenience, a button has been added above the table for downloading its contents in CSV format. The implementation for this export function is in a separate file, `CountryNetTableCsvExport.js`. Similar to the search, the download function is also carried out locally based on the data already retrieved from the server.

5.2.4 Charts

Charts have been added allowing to evaluate the data on a larger scale. Most importantly, a bar chart (figure 5.5) categorizing each AS detected in a country by combining two questions:

- Is the AS registered in this country?
- Did MaxMind find any associated IP addresses in this country?

There are four ways to answer both questions. If both questions are answered with no, the AS does not appear at all. For every other combination, there is a dedicated bar displaying the number of corresponding ASes.

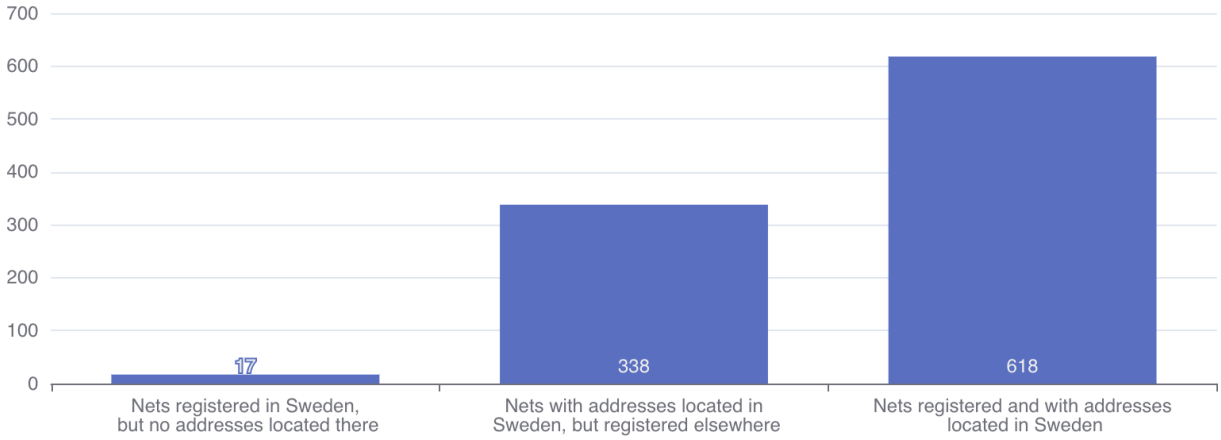


Figure 5.5: The bar chart shows which approach finds how many ASes in a country

This allows a very fast impression of how many ASes would also have been found only by looking at the country of registration (left and right bar) and how many are added by also considering the announced IP prefixes and MaxMind GeoIP (middle bar).

There are two more graph types showing the distribution of the size (in IP addresses) as a bar chart and the same information, after categorizing each AS by the percentage of addresses located in the registration country, in a cumulative distribution function (CDF) chart.

Furthermore, there are charts available at a global level. These can be reached via the button labeled *Charts* next to the search bar. Some of the country level charts also appear here. They are based on the same components which is made possible by implementing an optional filter for the selected country.

Individual charts are implemented as separate components in the directory `chart`. The charts displayed in the context of a country are grouped in the component `CountryCharts.js`, the charts displayed in the global view can be found in the component `Charts.js`.

The individual chart components like `IpsInNetBarChart.js` generally consist of some data mapping and grouping steps that derive the chart data from the overviews retrieved from the backend. These transformations could also have been implemented in the backend. While that approach would be more intuitive, it would also require storing the results in the backend and sending them over the network. In order to avoid excessive data consumption, it was decided to calculate chart data in the frontend instead.

More charts can be seen in chapter 6 (figures 6.2 and 6.3). There are additional charts available showing the number of nets whose percentage of IP addresses located within their country of registration are in a distinct range (below 20 %, below 40 % and so on) and a scatter chart plotting the number of addresses in an AS on a logarithmic scale against the percentage of them located within the country of registration. This is not particularly useful, but looks beautiful and was thus deemed to be deserving a spot in this paper.

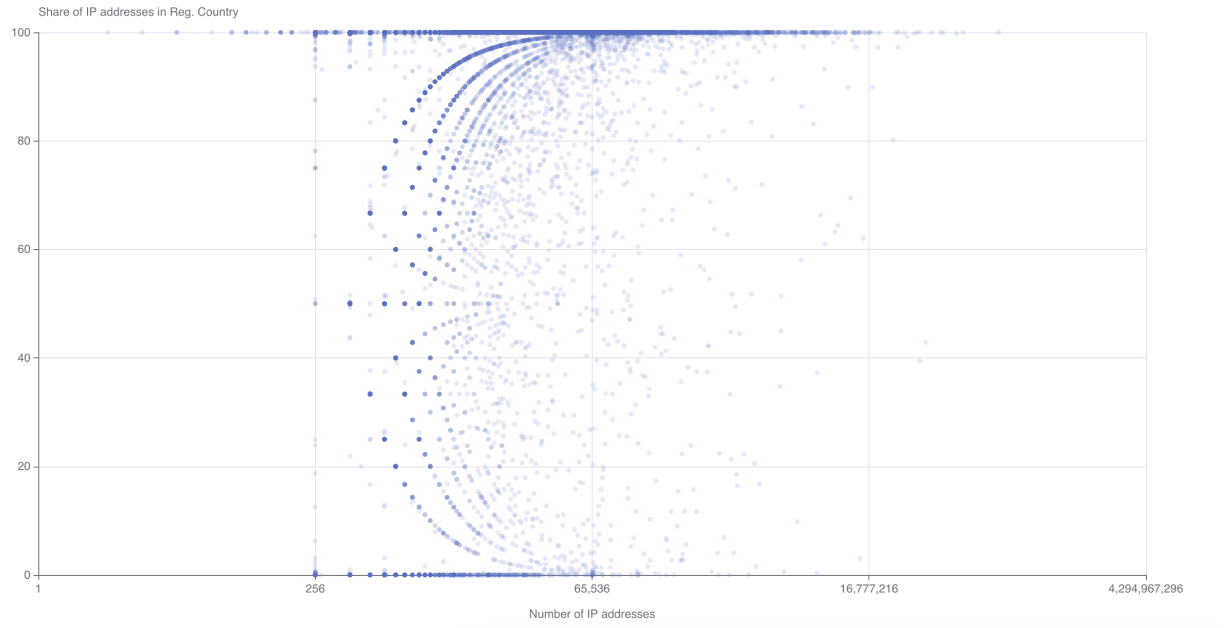


Figure 5.6: A scatter chart with each dot representing an AS with its share of IP addresses in its registration country (y) and its size as number of IP addresses (x)

6 Evaluation

Looking back, this project came to life quickly and successfully. With a conceptual and technical approach sketched out, implementation began and soon showed first results. A PeeringDB dump was downloaded and processed programmatically and then used as a data layer displaying peering facilities and organizations in an OpenStreetMap-based web application. From there, additional data sources were added and the data processing became increasingly complex. Features like the search bar, geolocating of Internet Protocol (IP) prefixes and country-centered autonomous system (AS) tables were added, allowing to visualize the produced dataset and help determining the answer to the research questions posed in section 1.3. A few weeks before finishing this work, the supervisors have already started using it productively, hosting the web application on a public server and using the results in professional data analyses.

However, it is still obligatory to have a closer look at the results in order to evaluate them. This is not trivial because there is no ground truth data available to compare the results with. Instead, the evaluation has to rely on the assessment of experienced professionals.

6.1 Deutsches Forschungsnetzwerk e.V. in New York?

There are outliers in the geographic locations provided by MaxMind GeoIP. For example, AS680 (Deutsches Forschungsnetzwerk e.V.) has a total of 7.8 million IP addresses, the vast majority of which is located in Germany. However, a single /24 IP prefix is located in New York. This outlier is suspicious and might hint at an actual error in MaxMind's database. Evaluating the quality of MaxMind's data is considered future work (section 7.1).

Note that this potential error was visible at the time of writing the previous paragraph. Two weeks later, after refreshing the data, the location in New York is gone. Instead, 100 % of the IP addresses announced by AS680 are located in Germany at the time of writing *this* paragraph. This shows that the Internet is dynamic and subject to frequent changes.

6.2 Overlapping prefix announcements

While examining the results of the Border Gateway Protocol (BGP) dump extraction (described in section 5.1.1.1), it became apparent that IP prefix announcements are not unique.

For example, AS7018 announces the prefixes `12.0.0.0/8` as well as `12.0.0.0/9`, although the latter is a subset of the former. This means that the first half of `12.0.0.0/8` is covered by both of the announcements. All charts involving numbers of IP addresses would have been falsified by this. If the two mentioned prefixes were the only prefixes announced by an AS, it would have been regarded as an AS with 25 million addresses ($2^{24} + 2^{23}$) while it actually only has 17 million (2^{24}).

Correcting this error was effortless utilizing Python's built-in library `ipaddress`. Its method `ipaddress.collapse_addresses` takes a list of IP prefixes and returns a consolidated list containing the least amount of IP prefixes necessary in order to cover all IP addresses contained in the input prefixes. In the previous example, the method would reduce the list (`12.0.0.0/8`, `12.0.0.0/9`) to (`12.0.0.0/8`), thus ensuring that each IP address is only contained exactly once.

In order to assess the difference that this change makes, a script was written that counts announced IP addresses, then consolidates the prefixes and counts the remaining, de-duplicated addresses. This revealed that a total of 20968 ASes announce overlapping IP prefixes. The

amount of IP addresses contained in multiple prefix announcements ranges from 1 of 26625 addresses (0.004 %) by AS39869 to 760320 of 891392 addresses (85.3 %) by AS37342. Globally, consolidating all IP prefix announcements for each AS removes 979 mio. of 4235 mio. addresses (23.11 %).

Note that this correction does not consider overlapping IP prefix announcements between different ASes. If AS1 announced `1.2.3.0/24` and AS2 announced `1.2.3.0/24` as well, the announcements would not be consolidated and the contained IP addresses would be counted for both of the ASes. Solving such cases is not trivial and considered future work (section 7.4).

6.3 Accuracy of MaxMind GeoIP

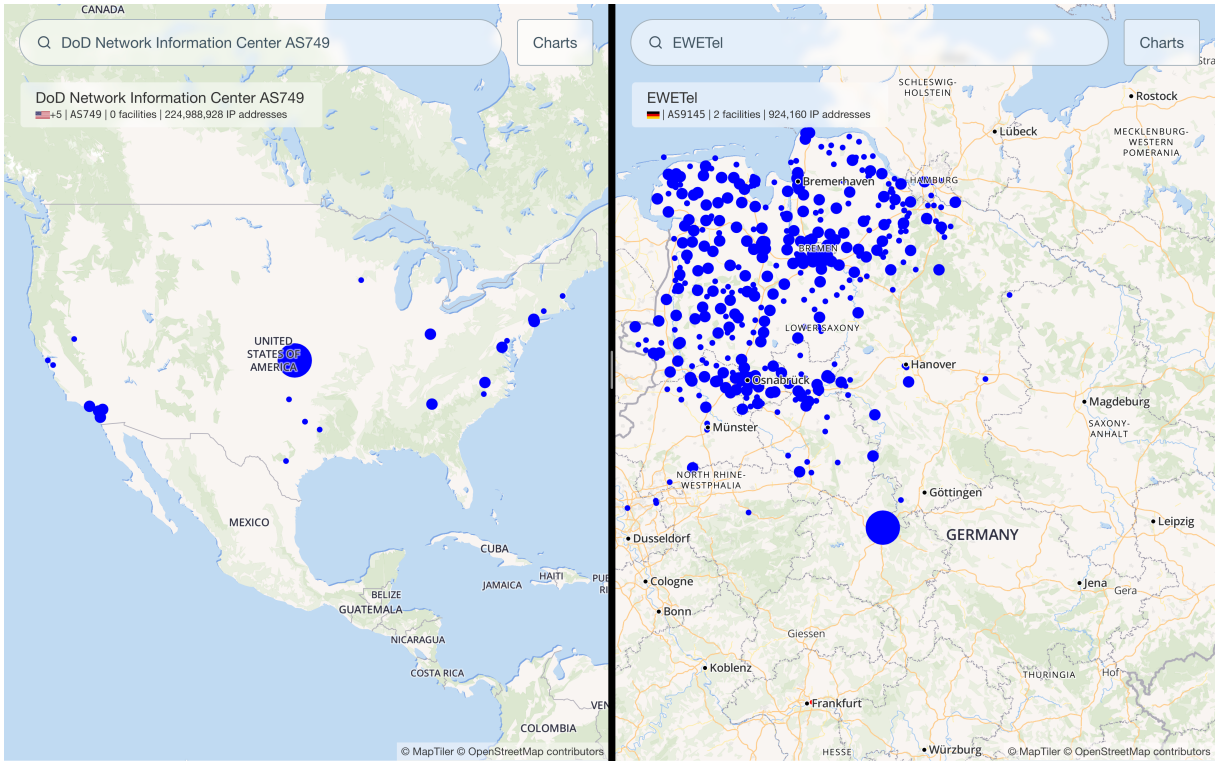


Figure 6.1: AS749 (225 million IP addresses in 30 locations) versus AS9145 (1 million IP addresses in 919 locations)

MaxMind's geographic locations are expected to be correct at country level [2]. In some cases this is all that can be expected. For instance, when selecting AS749 (DoD Network Information Center) – the single largest AS containing IP prefixes totaling around 224.99 million addresses – the map displays only 30 unique locations. The large circle in the middle of the US then represents 224.74 million addresses, which is the vast majority.

In other cases, such as AS9145 (EWETel), MaxMind draws a much more differentiated picture. The AS announces IP prefixes with a total of 0.9 million IP addresses. MaxMind GeoIP offers 919 unique locations for those addresses. Plotted on the map, the user can immediately identify a well-defined part of Northern Germany where the AS is using most of its IP addresses.

Figure 6.1 shows both cases in comparison.

6.4 How often are IP prefixes used outside the registration country?

The goal of this work is to find activities of ASes in countries outside their country of registration. It is thus sensible to inspect how many of their IP addresses are being used inside that country and how many are not since these are the cases that lead to finding more ASes in a country than just by considering their country of registration.

There is chart available showing the distribution of the share of IP addresses located in the registration country as cumulative distribution function (CDF) (figure 6.2). When holding the cursor over the graph, there appears a text explaining the hovered data point.

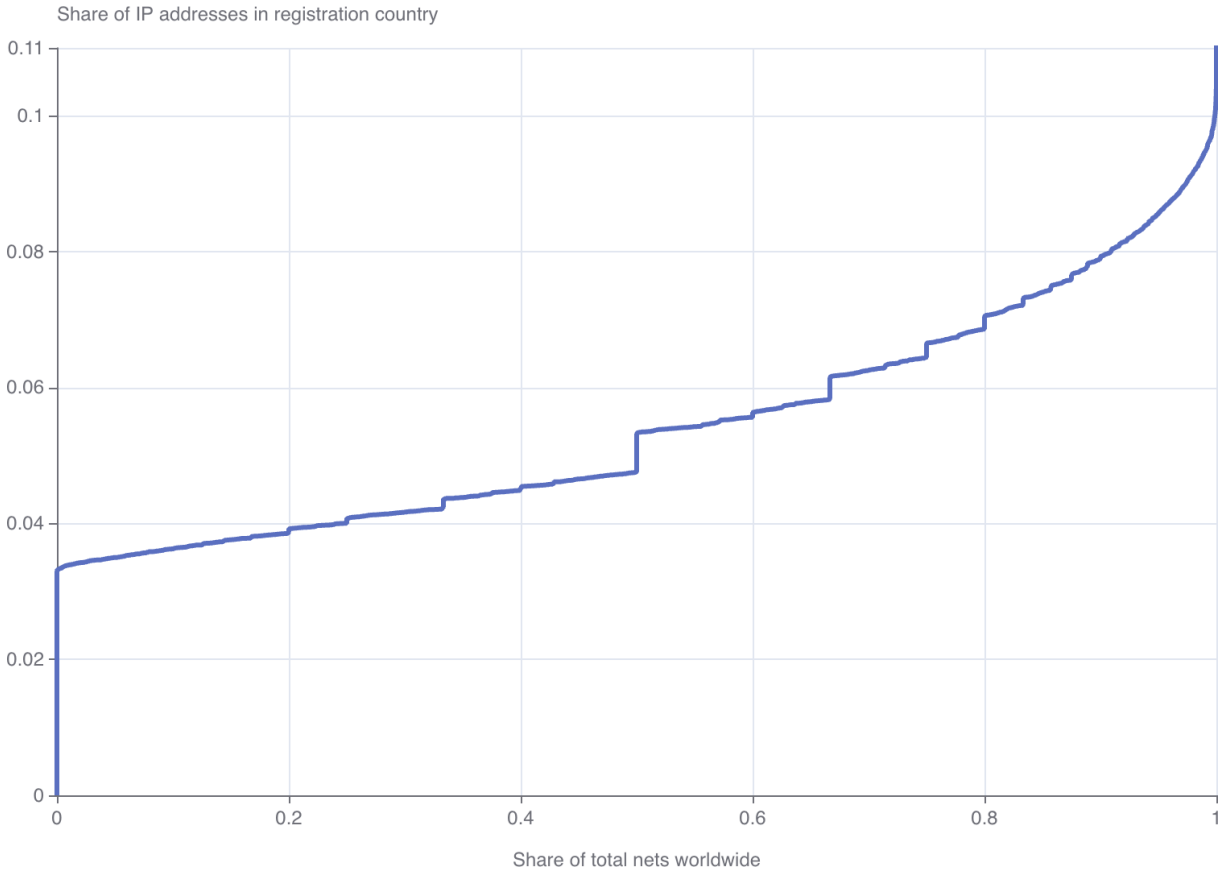


Figure 6.2: Share of IP addresses in registration country

First, note that the y-axis is cut off at $y = 0.11$. This is because for approximately 89 % of all ASes, their IP addresses were completely located inside their country of registration. For those ASes, interpreting their registration country as the sole country of operation would have yielded the same result as this work's approach.

The other 11 % of ASes visible in the chart, however, use a certain share of their IP addresses outside of the registration country. Indicated by the leftmost part of the graph, there is even 3.3 % of all ASes that have no addresses located in their registration country at all. Assuming those ASes were operating in their registration country would have been an error.

6.5 Closer look at Ethiopia

Ethiopia is a special case of a country with only 3 officially registered ASes. The advantage of taking additional data sources into account to find other ASes operating in a country becomes apparent here. The bar chart (figure 6.4) comparing the amount of ASes using different methods

shows that there are 17 ASes operating in Ethiopia while registered in another country.

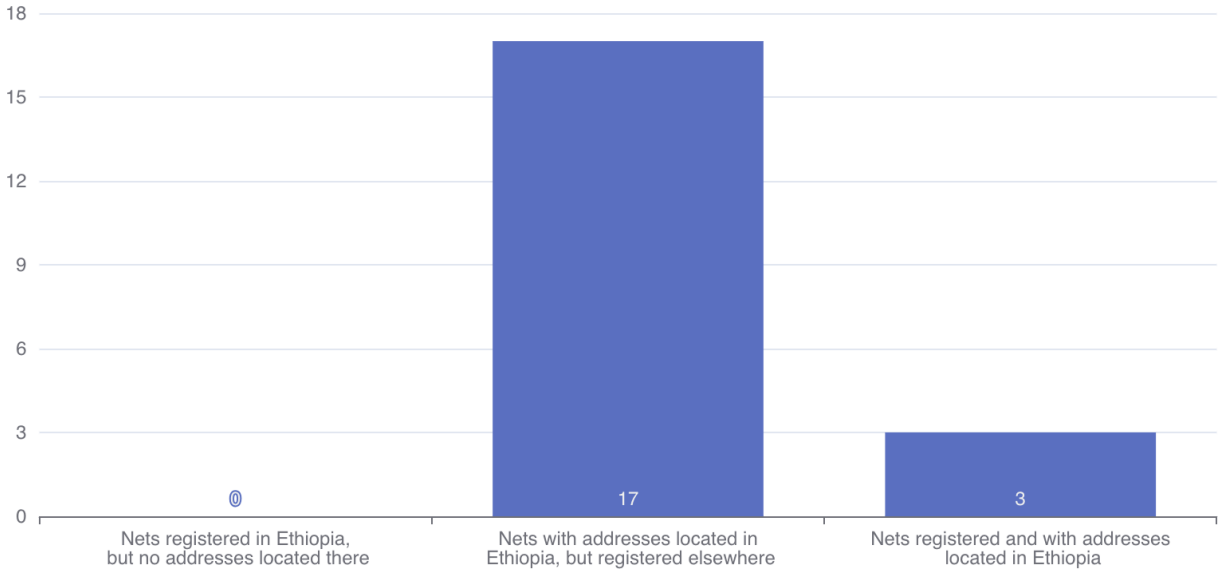


Figure 6.3: ASes in Ethiopia detected using different methods

Besides the 3 ASes registered in Ethiopia (AS24757, AS328988 and AS329265), the approach implemented in this work finds numerous additional ASes which are revealed in the AS table for Ethiopia as seen in figure 6.3.

Evaluating this information is best started in the rightmost column. The 17 additional ASes found in Ethiopia via MaxMind are, without exception, present in at least 67 other countries. This suggests that all of them can be considered a tier-1 AS. Among them are well-known names like Cloudflare, Google and Akamai who are expected to be present globally so that their presence in Ethiopia comes as no surprise.

6.6 Answering the research question

In section 1.3, two research question were posed: Can we utilize IP geolocation databases to get a clearer view of where ASes are operating? Given a country, how many ASes are operating there?

Evaluation made clear that involving an IP geolocation database does give deeper insight to where ASes are operating. For any given country, the approach implemented in this work shows a significant amount of ASes that have a presence there while it is registered in another country. Only considering the country of registration would leave those ASes unseen.

So the answer is clear: Yes, IP geolocation databases can be used to get a clearer view of where ASes are operating.

ASN	AS name	IPs	IPs in Reg. Country	Reg. Country	Present in Country
AS13335	Cloudflare	1,689,341	20.6 %		+248
AS15169	Google LLC	9,001,472	64.4 %		+88
AS16276	OVHcloud	4,305,920	63.4 %		+248
AS18779	Energy Group Networks	1,414,400	91.0 %		+67
AS20940	Akamai Technologies	13,915,840	3.1 %		+236
AS23026	SETEC ASTRONOMY	512	0.0 %		+1
AS24757	Ethio Telecom	296,960	99.7 %		+1
AS29286	Eutelsat - Skylogic	162,304	32.6 %		+68
AS34800	Securebit [Europe]	255	1.6 %		+248
AS36183	Unknown	141,565	36.8 %		+236
AS36492	Unknown	48,382	68.0 %		+172
AS37518	Fiber Gride\	1,114,112	4.1 %		+195
AS44444	Forcepoint Cloud	48,640	7.4 %		+136
AS54113	Fastly, Inc.	329,728	68.0 %		+236
AS55256	Netskope	86,272	53.9 %		+172
AS58057	Securebit AG	12,800	80.0 %		+248
AS198605	AVAST Software s.r.o.	20,736	34.5 %		+192
AS203061	IT Proximus	56,064	0.0 %		+119
AS328988	Unknown	2,048	100.0 %		
AS329265	Unknown	256	100.0 %		

Figure 6.4: AS table for Ethiopia

7 Future Work

This work may serve as a starting point for various future work.

7.1 Detecting errors in MaxMind GeoIP

The data in MaxMind GeoIP varies in accuracy. While its geographic locations have been shown to be mostly correct at least at country level [2], it is still possible to find errors in the database. Potential errors can also be seen on the map in this work. For example, one outlier discussed in section 6.1 used to hint at a possible error in the database. A systematic analysis of such outliers can give more insight into how accurate and reliable the data in MaxMind GeoIP are.

7.2 Considering other IP geolocation databases

In order to find the aforementioned outliers and to generally improve results, there are other products like MaxMind GeoIP at hand. Taking into account and comparing different Internet Protocol (IP) geolocation databases at the same time increases confidence in the geolocation results and allows assessing the quality of each of the used databases.

7.3 Considering peering facilities

Peering facilities were in the first dataset produced in the earliest starting phase of this project. While they are being displayed on the map and give a hint towards possible activity of an autonomous system (AS) visually, they are currently excluded from answering the research questions (section 1.3).

With respect to the three categories defined by combining two questions (Registered in country?, Present in country according to MaxMind?) as visualized in the bar chart 5.5), a third question could be added, asking: Is the AS connected to a peering facility in the country, according to PeeringDB? Instead of three categories, this would yield seven possible categories of AS presence in a country. One where an AS is registered in the country, and present according to MaxMind, and peering with facilities in the country. And six additional categories derived from the other possible combinations of answers to those questions.

Considering peering facilities like that could give an even more detailed answer to the research questions.

7.4 Solving overlapping IP prefix announcements

An assumption was made that IP prefix announcements are unique. In section 6.2 it was already described that this is not the case because ASes can announce overlapping prefixes. This could be solved by consolidating the announcements per AS.

However, it is also possible that multiple ASes announce the same IP prefix or different prefixes with one being a subset of the other. Some ASes announce wrong prefixes that do not belong to them. Furthermore, some companies own several ASes and announce their prefixes not only from one, but multiple ASes. Considering such AS groups is complex and thus outside the scope of this work.

7.5 Optimizing user interface

The implemented user interface offers room for improvement. While the AS table (section 5.2.3) can be grasped for small countries like Ethiopia (section 6.5), larger tables can become confusing and difficult to work with. It could help to add functionality for sorting and filtering the table by different criteria.

7.6 Optimizing network traffic

In its current state, lots of data is transmitted between the backend and the frontend. Especially the file `nets.json` (section 5.1.3) can be considered a heavyweight at around 20 megabytes. For private applications, this can be neglected. In public use however, this behavior could result in an enormous use of resources. There are various options to optimize network traffic throughout the application.

Utilizing compression to reduce the size of transmitted data has a great potential. The JavaScript Object Notation (JSON) files are plain text with lots of white space and repetitive content. One way close at hand to implement compression is enabling server-side compression.

Taking this a step further, choosing a different format might reduce the file sizes even more efficiently. A binary format storing numbers (like autonomous system numbers (ASNs)) as actual longs instead of strings has the potential to save a lot of bandwidth.

Another approach to save on network traffic would be implementing a cache in the frontend so that multiple requests to the same resource only have to be served by the backend once. This is especially feasible for this application as all data served by the backend is static.

7.7 Optimizing the map style

The current implementation uses a standard style for the map in the frontend. It includes a lot of detail that might be relevant to users looking for specific information in their area, including buildings, footways, points of interest and more. For visualizing the results of this work, these details are at least irrelevant, if not even distracting. Choosing a lighter style for the map that includes only relevant information can make the web application more accessible and help the user focus.

7.8 Making the web application available to the public

Publishing the web application requires a few extra steps. A web server is needed serving the content from the backend as well as the built web application. The frontend needs to be aware which Uniform Resource Locator (URL) it has to use to reach the backend. A pipeline should be set up to execute all the backend script in order to refresh the data on a regular basis.

The map relies on the usage of map tiles served by the third-party service MapTiler. For these tiles there is a quota for free use, which will be exceeded when the use of the application increases. The MapTiler membership can be upgraded for a fee to provide more tiles. Alternatively, a standalone server for map tiles can be installed, eliminating the need for paying a third party. This is notably practical when the map is not required to display a lot of features in low zoom levels (see section 7.7) which significantly impacts the amount of data necessary for hosting a map tile server.

As described in chapter 6, a semi-public instance of the application has already been deployed.

7.9 IPv6 support

This work considers Internet Protocol Version 4 (IPv4) only. Its successor, Internet Protocol Version 6 (IPv6), has been around for tens of years. Given the shortage of IPv4 addresses, IPv6 becomes increasingly important.

By also considering IPv6 prefix announcements, additional links between ASes and countries might be found in IP geolocation databases. IPv6 prefix announcements are already present in Border Gateway Protocol (BGP) route collector dumps and parsed by the backend (section 5.1.1.1), but not processed further. Supporting IPv6 in this work requires adjustments to the code and data structures both in the backend and in the frontend.

8 Conclusion

In the summer of 2023, the author of this work set out to find a topic for his bachelor's thesis. Upon reading about the topic of geolocating autonomous systems advertised by the Distributed Systems Group, he first had to look up the term autonomous system.

Some time has passed since then and he has gathered some knowledge in the field. He learned about autonomous systems, Border Gateway Protocol (BGP) routing and route collectors, the Multi-Threaded Routing Toolkit (MRT) format, peering facilities and Internet exchange points (IXPs), various tiers of autonomous systems (ASes) and how and why they connect to each other, the Internet Assigned Numbers Authority (IANA) and the five regional Internet registries (RIRs). PeeringDB was introduced, a Python script was written, a React app was set up and suddenly, there were little dots on a map, representing peering facilities extracted from a PeeringDB data dump.

Things got interesting from there. Having learned about ASes and related topics and thus understanding the implications of the research question (section 1.3), the author began envisioning a data processing pipeline connecting data from various sources. Keeping efficiency, readability, usability and extendability in mind, he expanded the backend continuously, becoming more fluent in Python along the way. The frontend was developed in parallel.

Several data sources (section 4.1) were added to the backend. Common data fields were identified and utilized in order to glue the pieces together (section 5.1.2). An intuitive frontend with a map and some charts were implemented to make interpreting the results and exploring the generated data as easy and fun as possible (section 5.2).

The resulting application can answer the research question (section 1.3) clearly: Yes, Internet Protocol (IP) geolocation databases can be used to get a clearer view of where ASes are operating! This is implicitly confirmed by people working in the industry who already started using the web application as a tool in their everyday work even before the thesis has been finished.

Besides, this work is merely a starting point for many further additions and improvements to come (chapter 7). Multiple IP geolocation databases are waiting to be connected, compared and integrated into the backend to improve the results even further (section 7.2). Various improvements can be made to the backend, the frontend and their connection, including finding a solution for overlapping IP prefix announcements between ASes (section 7.4) and adding support for the Internet Protocol Version 6 (IPv6).

The author is looking forward to watching the project grow in the future and wishes everyone involved a good time!

Bibliography

[1] W.B. Norton. *The Internet Peering Playbook: Connecting to the Core of the Internet*. DrPeering Press, 2011. ISBN: 9781937451004. URL: https://books.google.de/books?id=rkDz6fvX_XkC.

[2] Ingmar Poese et al. "IP geolocation databases: unreliable?" In: *SIGCOMM Comput. Commun. Rev.* 41.2 (Apr. 2011), pp. 53–56. ISSN: 0146-4833. DOI: 10.1145/1971162.1971171. URL: https://doi.org/10.1145/1971162.1971171.

Statutory Declaration

The author declares that he has written the thesis at hand independently, without outside help and without the use of any other but the listed sources. Thoughts taken directly or indirectly from external sources (including electronic sources) are marked accordingly without exception. Sources used verbatim and contentual were quoted according to the recognised rules for scientific working. This thesis has not been submitted in the same or similar form, not even partially, within the scope of a different examination.

Thus far it also has not been publicised yet.

I herewith agree that the thesis will be examined for plagiarism with the help of a plagiarism-detection service.

Place, Date

Signature of the author