as2org+ : Enriching AS-to-Organization Mappings with PeeringDB

Augusto Arturi[1], Esteban Carisimo[2], and Fabián E. Bustamante[2]

[1] Universidad de Buenos Aires {aarturi}@fi.uba.ar
[2] Northwestern University {esteban.carisimo, fabianb}@cs.northwestern.edu

Abstract. An organization-level topology of the Internet is a valuable resource with uses that range from the study of organizations' footprints and Internet centralization trends, to analysis of the dynamics of the Internet's corporate structures as result of (de)mergers and acquisitions. Current approaches to infer this topology rely exclusively on WHOIS databases and are thus impacted by its limitations, including errors and outdated data. We argue that a collaborative, operator-oriented database such as PeeringDB can bring a complementary perspective from the legally-bounded information available in WHOIS records. We present *as2org+* , a new framework that leverages self-reported information available on PeeringDB to boost the state-of-the-art WHOIS-based methodologies. We discuss the challenges and opportunities with using PeeringDB records for AS-to-organization mappings, present the design of *as2org+* and demonstrate its value identifying companies operating in multiple continents and mergers and acquisitions over a five-year period.

1 Introduction

An understanding of the Internet topology, its properties and their evolution, is critical to a number of research questions from routing [21,40] and application performance [50,19,49] to network security [53,52,13], Internet resilience [1,17,26,56,43], and Internet governance [28,34,39].

As a network of networks, the Internet is composed of over 73,000 Autonomous Systems (ASes) that cooperate via the Border Gateway Protocol (BGP) to exchange routing information and obtain global reachability. The connections between ASes are shaped by the business contracts between the organizations that manage them, and that define the economics and technical aspects of exchanged traffic.

Many Internet studies over the years have focused on an Internet topology defined by the ASes and their relationships, building on different heuristics to infer AS relationships from publicly available BGP routing data [31,40,16]. AS relationships fall into three broad classes: customer-provider, settlement-free peering and siblings. In a customer-provider relationship, a customer AS pays a provider for reachability to/from the rest of the Internet. In a settlement-free peering, two ASes agree to exchange traffic destined to networks they or their customers own, without an associated fee. A sibling relationship exists between

distinct ASes that are owned by the same organization and can exchange traffic without any cost or routing restrictions. Although seemingly straightforward, this approach ignores the relation that exists between the AS-level topology, the organizations that make up the Internet, and the rich semantic content that is key to its understanding and proper use in a range of analysis, from characterizing trends towards Internet centralization [42,27,35,32] to understanding the impact of business disputes [15], public policies [28] and legal actions [57].

In their seminal work, Cai et al. [57] define the problem of AS-to-organization mapping and present methods to generate an organization-level view of the AS ecosystem. We adopt their definition of organization as *an entity which has control over itself and is not a subsidiary of any other organization*. Organizations may include multiple ASes as a result of company merges and acquisition or to facilitate other, more complex arrangements such as different business units, or alternative routing policies for different parts of their network. An organization-level topology, thus, clusters together entities sharing common business decisions, showing two organizations as connected if there exists a relationship between at least one of their affiliated ASes.

The state-of-the-art AS-to-Organization mapping method, `AS2Org` [57], extracts organization information from AS registration data available on WHOIS records to identify ASes under the same management. WHOIS records, however, are known to contain inaccurate and outdated information which impact the accuracy of the inferred Internet organization-level topology (§2), ranging from the simply out-of-date records resulting from mergers and acquisitions or incongruence between commercial names and registration data, to the challenges that come from capturing the different approaches that large corporations use to structure their organizations (e.g., having independent organizations for their country-level subsidiaries).

We argue that *self-reported* information available on PeeringDB can be leveraged to boost AS-to-Organization mappings and address many of these challenges. PeeringDB (PDB) is an online open database established in 2004 to assist peering coordinators identifying potential peers and peering locations. AS operators voluntarily provide information about their networks, such as peering policies, traffic volumes and presence at various geographic locations. In the past decade, PDB has become the *de facto* public profile of Internet networks. There are a number of factors that explain the popularity of PDB, including the fact that main cloud and content providers request its peers to be listed on PDB to establish peering relationships [41,18,25]. Despite participation in PDB being voluntary with no mechanism to verify the accuracy of reported information, prior work has shown it to be mostly correct [38] and several studies have relied on it to infer the size of Hypergiants [5] and determine the relevance of peering facilities [24].

We present $as2org+$, a new framework that leverages PDB data to improve on the state-of-the-art AS-to-Organization mapping methodology. $as2org+$ builds on the insight that a collaborative operator-oriented database could bring a com-

plementary perspective to the legally-bounded information available in WHOIS records.

We face a number of challenges (§2) in leveraging PDB data including the operators' use of non-standard fields to communicate siblings, even when PDB provides an Organization Identifier (OrgID), and the use of loosely structured formats meant to be read by humans. We evaluate the contribution of PDB-based inferencing (§6) and its contribution to the AS-to-organization mapping problem (§7). We demonstrate the value of *as2org+* to derive more complete organizational structures of large transit providers (§7.2) and Hypergiants (§7.3).

In sum, our work makes the following contributions:

- We propose the use of PDB as a valuable source to enhance AS-to-Organization mappings, and present a methodology to extract self-reported siblings embedded in PDB records.
- We present *as2org+* , a new framework for AS-to-organization mapping that combines PDB-based inferences with the current WHOIS-based approach to enhance organization-level topology
- We evaluate *as2org+* contributions to the Organization level topology and discover that it provides a more complete representation of large transit networks (26 networks in CAIDA's AS-RANK TOP100) and Hypergiants, for example grouping together different subsidiaries and business units of Google (Google, Google Cloud Services and Google Fiber) and Akamai (Akamai, Prolexic and Linode).
- We apply *as2org+* to a five-year dataset and find that it enables clustering together networks as a result of mergers and acquisitions that are not visible in WHOIS-based datasets, such as GTT (AS3257) acquisition of KPN (AS286) or CenturyLink (AS209) acquisition of Level3 (AS3356).
- We contrast AS2Org's RIR-level scope with PDB's geographically-unconstrained organizations and find that *as2org+* is able to group together companies operating in multiple continents, for instance *as2org+* groups in the same cluster Yahoo's subsidiaries in the US, UK and Japan.
- We make *as2org+* [3] available to the community.

This work does not raise any ethical issues.

2 Motivation and Challenges

A more complete Organization-level topology would be a valuable resource for a wide range of disciplines. Improved AS-to-Organization mappings contribute to a better representation of private or state-owned organizations' footprint [9]. Having a more complete representation of state-owned organizations could help us to understand governments' engagements in the Internet, as a business activity, domestically and abroad. This could also be a valuable resource to identify market

[3] *as2org+* can be found at: https://github.com/NU-AquaLab/as2orgplus

concentration of Internet resources at an organization level. Improved AS-to-Organization mappings can also help better understand the constant reshaping of the corporate structure of the Internet as result of mergers, de-mergers and acquisitions, such as CenturyLink's acquisition of Level3 [12] (now rebranded as LUMEN), the merger between T-Mobile and Sprint [54] or the recent de-merger between Telia's transit network (Telia Carrier, now rebranded Arelion) from the rest of the company [14].

The state-of-the-art AS-to-Organization mapping technique bases its inferences exclusively on WHOIS data. Despite providing a complete coverage and valuable information of the allocated resources, WHOIS databases have several limitations (from errors and outdated data, to entries with unstructured text formats) that impact the accuracy and coverage of methods reliant on this source. In the next paragraphs we describe some limitations and the challenges of using WHOIS data to identify networks under the same ownership.

Corporate business segmentation: Large corporations use different approaches to structure their organizations into *separate legal entities* running business units that include departments, divisions and subsidiaries. The lack of common practices, as well as the size and complexity of these organizations, create challenges to fully capture the business structure of these companies. Key to understanding this challenge is that network resource allocations are given to a single legal entity considering each resource holder as an independent organization. Internet access providers operating in multiple countries (*e.g.,* Orange, Deutsche Telekom or Claro) are likely to be segmented in multiple subsidiaries (a separate legal entity) with resources specifically allocated for operating each of them. Network segmentation of multinational Internet providers vary from company to company, but most approaches include a (nearly) per-country subsidiary with their own network resources. Other large corporations run a diverse portfolio of Internet businesses (*e.g.,* Internet access, content delivery) and are likely to have different companies and/or networks (and therefore network resources) for each business. This is the case, for example, of Google with Google Fiber as an Internet access provider. Claro – a mobile carrier with an extensive footprint across Latin America– offers another example. Each of Claro's country-level subsidiaries is registered as an independent organization (Claro Argentina (AS19037: `AR-CCTI1-LACNIC`), Claro Chile (AS27995: `CL-CCSA39-LACNIC`)). These business dynamics and practices are poorly captured by WHOIS records where there are no clear relationships between different assets of the same conglomerate.

(Mis)communications from resource holders: Despite contractual obligations requiring resource holders to maintain information up to date[4], resource holders are unlikely to contact the RIR for issues that are not regarding to renew or upgrade allocations. As a result of the lack of communication, many delegation records do not properly capture the status of the organization. In recent years, ARIN acknowledged that mergers and acquisitions create challenges

[4] Legacy resources [4] — allocations preceding the creation of RIRs — are also subject to different regulations [46]

to organizations to coordinate all the allocated resources to report the same information [46].

RIR-level allocations: Corporates controlling subsidiaries in different regions are going to be treated as separate organizations for allocation purposes since RIRs' allocation policies require organizations to be an *"active business entity legally formed within the RIR service region"* [3]. The RIR-level scope of organizations included in WHOIS data limits our ability to fully capture organizations of corporations with presence in different RIRs. For example, the ASes of French-based Orange and its subsidiary in Cameroon are identified by different OrgIDs (AS5511: `ORG-FT2-RIPE`, AS36912: `ORG-OCS1-AFRINIC`).

Data accuracy and formats: Limitations and inaccuracies in the process of data collection and data presentation of WHOIS records limits and hinders WHOIS-based methodologies. WHOIS data schemas are not homogeneous across RIRs (and NIRs too) where syntaxis (field names), semantics (field content) and number of elements vary across different regions. Another methodological limitation is that WHOIS records are (mostly or exclusively) accessible through the WHOIS protocol and retrieved data is returned as loosely cohesive plain text [36]. In quarterly released AS2Org mappings, CAIDA homogenizes and structures the WHOIS data [8]. Despite these efforts to improve the quality of the data product, registration and resource allocation involves human intervention and these forms are prone to errors.

Incongruence between commercial names and registration data. Corporations could have homogeneous brand names across subsidiaries but WHOIS databases may not capture that homogeneity since resource holders tend to fill up the registration name (`OrgName field`) with the company's legal name, which may differ from commercial names or brand names. As an example, Colombia's state-owned Internexa [9] operates in Argentina the AS262195, however, LACNIC's WHOIS reports the owner's name to be *Transamerican Telecomunication S.A.* This incongruence between commercial and registration names present barriers for analysis that uses WHOIS data to identify text similarities.

These are just examples of the limitations and challenges faced by the state-of-the-art AS-to-Organization mapping approach and partially motivate our work. Despite WHOIS-based AS-to-Organization mappings being incomplete, we believe this is a valuable data source that could be enhanced with organizational data obtained from alternative sources, such as PeeringDB.

3 Challenges and Opportunities with PeeringDB

While we argue that the growing popularity and use of PeeringDB can offer a complementary perspective to traditional WHOIS-based approaches, its use is not without challenges. For instance, the database is voluntarily and does not provide complete or uniform coverage across regions which could potentially introduce biases in AS-to-Organization mappings. We also find that despite PDB providing an Organization Identifier (OrgID), operators sometimes rely on other fields to communicate siblings, and that in some cases those siblings do not even

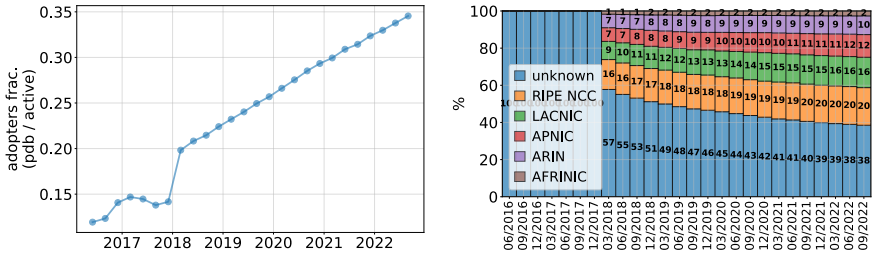
have presence on PeeringDB (*e.g.,* Tigo-AS262206 reports AS26617 as a sibling in text fields but this network is not registered in PDB). We also find that this information is often loosely structured as it is intended to be read by human operators.

In the following paragraphs we discuss some additional challenges with using PDB to identify ASes belonging to an organization, and potential approaches to take advantage of its rich information.

3.1 PDB for AS2Orgs mapping

A non-exhaustive set of limitations of PDB that could impact sibling inferences include its relatively limited coverage, bias in its adoption by operators, and potential issues of completeness and correctness of the database. We briefly discuss each of these limitations in the following paragraphs.

Limited coverage: Despite PDB adoption being steadily growing, as shown in Figure 1a, this database presents a limited visibility of the AS ecosystem where only ≈34.5% (25,767 / 74,583) of active networks[5] has registered in PDB. However, the adoption seems to be skewed towards prominent networks which we expect to have more complex organizational structures, such as large transit networks where 100% and 93.8% of CAIDA's AS-RANK [7] TOP100 and TOP1000 are registered in PDB. We also expect this number to keep growing due to some Hypergiants (HGs) requiring PDB profiles to establish peering sessions with peers [41,18,25] and government and IXP initiatives encouraging and helping local ASes to join PDB [45,6].



(a) (b)

Fig. 1. PeeringDB adoption as fraction of active ASes (left) and per region (right).

Geographic bias: PDB adoption rate may vary across countries depending on peering incentives (*e.g.,* local presence of HGs), consolidated peering ecosystems (*e.g.,* presence of large IXPs), common communication practices among

[5] We refer as active ASes to Autonomous System Numbers visible in BGP routing tables.

local operators, *etc.*. A previous study conducted in 2013 found that RIPE is over-represented across the networks registered at PDB, while AFRINIC and LACNIC registered networks had a small footprint [38]. We examine the *self-reported* country information of the organizations registered in PDB, as shown in Figure 1b, and found a growing adoption at a RIR level in APNIC, and more remarkably, in LACNIC regions. Despite lacking *self-reported* country information for a fraction of the records, we observe that PDB adoption is not confined to specific regions giving us visibility of all regions.

Completeness, correctness and use of fields: The fact that PeeringDB is an open and voluntary database raises questions about the accuracy and authenticity of the included data. PeeringDB uses a series of mechanics with authoritative data sources (WHOIS, RDAP [2,30], *etc.*) to authenticate that the data source is legitimate [47]. On the other hand, the accuracy of PDB records is encouraged by the fact that inflated statistics could compromise peering agreements or harm the reputation of the networks [38].

3.2 Opportunities using PeeringDB

We now analyze the PDB data schema (version 2) to identify elements that could potentially inform *siblings*. We investigate whether being an operation-oriented database could bring a different perspective to the sibling inference problem compared to WHOIS information, which refers as an organization to legal entities in a specific RIR. We identify two main ways organizations use to communicate the set of ASes under their management: *(i)* use of native features of PDB data schema (`org` data structure) *(ii)* custom use of plain text fields (*e.g.,* `aka`, `notes`).

Among the several data entities available in PDB, we focus on those that are more relevant for this work: organization (`org`) and network (`net`). The data entity `org` describes organizations with fields such as `name`, `also known as (aka)`, `website`, `address`, `country`, *etc.*. However, the most important attribute of these entities is the network field which is a list of network identifiers referring to `net` entries administered by the organization The data entity `net` describes ASes with fields such as `name`, `also known as (aka)`, `network type`, several network attributes (*e.g.,* number of IPv4 prefixes), peering, and more importantly the `organization` field referring to the organization this network belongs to. By combining both data entities using the list of bidirectional network/organization identifiers, we can directly generate AS-to-Organization mappings.


```
{"meta":
 {"generated": 1601614591.736},
  "data": [
   {
    "asn": 4436,
    "website":"http://www.gtt.net",
```

```
7       "notes": "nLayer / AS4436 has been acquired by GTT
            Communications / AS3257 and is no longer directly
            peering.Please refer all peering related inquiries to
             peering [at] gtt [dot] net.",
8       "org_id": 8897,
9       "policy_url": "http://www.gtt.net/peering/",
10      "aka": "Formerly known as nLayer Communications",
11  }}}
```

Listing 1.1. Example of the `net` entry for AS4436 in the PDB snapshot of Oct. 2020.

We further investigate whether the content reported in fields of `org` and `net` could provide some information of other ASes operated by the same organization. Listing 1.1 shows an example of some fields in the `net` entry of AS4436 (nLayer) to explain how these fields could provide hints about siblings. In this specific case, nLayer was acquired by GTT in 2012 [55] and this information is available in the `notes`, where both nLayer (AS4436) and GTT (AS3257) ASNs are included. Ten year later, both networks are under different organizations in WHOIS records. The use of `aka` in this example is informative but insufficient to obtain a cluster with ASNs of both networks. In Appendix A we include an example of a `net` entry in which operators used the field `aka` to report ASes under the same management.

4 Methodology

In this section we describe our methodology to extract siblings from PDB records. This methodology offers two types of sibling inferences – *conservative* or *aggressive* – depending on the confidence level of the obtained results. Figure 2 illustrates the pipeline of the methodology before consolidation with the AS2Org's inferences. The *aggressive* part of the methodology consists of four main stages each: *(i)* feature extraction *(ii)* filters *(iii)* inspection and *(iv)* data consolidation.

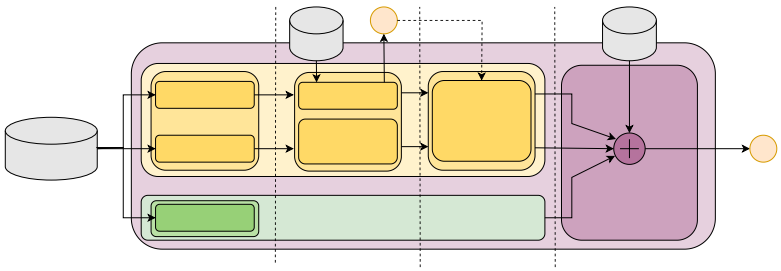


Fig. 2. Diagram of the *as2org+* framework.

The *conservative* approach only uses `org_id` present in the `net` data entity and it does not apply any heuristic to infer siblings. On the other hand, the *aggressive* approach applies heuristics to extract self-reported *siblings* ASNs embedded in either the `aka` field or the `notes` field (or both). In this approach, we create *candidate groups of ASes under the same administration* as an output and later apply filters to improve confidence. The *conservative* approach is a *zero-risk approach* since PDB applies mechanisms to authenticate the ownership of a network resource (see §3.1), preventing two non-sibling ASNs from being identified by the same `org_id` . The *aggressive* approach could potentially include numbers that are not *ASNs under the same managements*, though, those false positives are mitigated by the design of our framework. We give users full control of the combination of these approaches where they can choose any combination of features. Next, we describe the implementation of our heuristics.

Before starting our process, we sanitize the data from the selected inputs and normalize the text (*e.g.,* case).

4.1 Feature extraction

Our PDB-based inference methodology uses three fields of PDB's net entity, the `org_id` field in the conservative approach, and `notes` and `aka` fields in the aggressive approach. In this stage, the conservative approach uses the `org_id` to group together all ASNs were registered by the same organization while the aggressive approach combines regular expressions (`regexes`) to extract groups of ASNs embedded in these fields. Next, we describe the rules applied to extract *self-reported siblings* embedded in these fields.

org_id. This feature extraction mechanism leverages the native `org_id` field in the PDB data schema to group together all ASes registered by the same organization.

aka. For this field, the framework applies a single regular expression that extracts numbers with 4 to 8 digits to generate the list of *candidate siblings*. We suspect that length constraints of this field (limited to 255 characters [48]) discourage operators from rich semantic statements and hence, sibling ASNs (sometimes along with AS names) are directly reported. In Appendix B we show a few examples of how operators report their networks in the `aka` field as well as the output of this regex. We acknowledge that this extraction method can result in wrong inferences. This rule is not capable of inferring *candidate siblings* ranging between AS1 and AS999. However, this impact is limited to missing at most 1% of the *siblings* since at the time of this submission more than 100,000 [44] have already been allocated. To be more specific, this rule lacks the semantic context of the numbers extracted, potentially leading to infer as *candidate sibling* strings such as dates and phone numbers. We apply custom filters (§4.2) to mitigate the presence of spurious numbers.

notes. We develop 37 `regexes` to extract *candidate lists of sibling ASes* embedded in different semantic contexts in the `notes` field. This is a data rich field (it is an unlimited plain text field [48]) that allows operators to include details that do not fit well in any other field, including detailed descriptions or

Table 1. Examples of simple feature extraction rules for the notes

ASN	Input	`regex`	output
21202	*AS21202 IN THE NORDIC REGION*	`AS[0-9]+`	[21202]
5462	*THIS AS IS BEING MIGRATED TO AS:5089*	`AS:[0-9]+`	[5089]
55818	*OPERATING 2 ASNS (55818 AND 45147)*	`ASNS.*`	[55818,45147]
35742	*THIS AS WILL BE MERGED SOON INTO AS 43646.*	`AS [0-9]+`	[43646]
10158	*HAS 6 ORIGIN ASS: 10158, 45991, 38678, 9764, 7625, 38099*	`ASS:.*`	[10158, 45991, 38678, 9764, 7625, 38099]
54113	*AUTONOMOUS SYSTEM (AS) 54113*	`[(]AS[)] [0-9]+`	[54113]
58715	*IIG(ASN-58715) & ISP(ASN-63969)*	`ASN-[0-9]+`	[58715, 63969]

specific requirements and procedures to peer with the network. The flexibility of the field and diversity of data reported (siblings, peering policy, capacities, NOC hours, *etc.*) sets challenges to identify a *candidate list of siblings*. Moreover, there is no convention to report these features, and the text structure can vary significantly as these messages are meant to be read by human operators.

We categorize the 37 `regexes` into two groups: simple rules (21) and complex rules (16). Simple rules aim to extract ASN from simple patterns that are used to refer to ASes using prefixes such as *AS, ASN, ASNS, ASS* and *ASES*, as it is shown in Table 1. Complex rules aim to extract ASNs from notes using more complex semantic expressions. We search for common phrases used (with a maximum of three words) to report ASes under the same management, including *also manages*, *we administered*, *merging*, as it is shown in Table 2. Due to a lack of a common structure, we consider *candidate siblings* to all numbers after this template phrase. This decision comes at the risk of including numbers unrelated to ASNs, such as addresses, RFC numbers, ISO standards and others. We also acknowledge that complex rules are only capable of extracting *siblings* of records written in English. In our implementation users can select using simple, complex or both rules for sibling inferences. In Section 6.3 we evaluate the contribution of each of these rules.

4.2 Filters

To remove numbers misinterpreted as ASN in the previous stages, include a filtering layer in the aggressive approach pipeline. We focus on filtering out errors coming from two sources, *(i) spurious numbers* (*e.g.,* phone numbers, addresses, years, RFC numbers *etc.*), and *(ii)* reported-but-not-sibling ASNs. To mitigate these false positive inferences, we develop two filters: *(i)* a spurious-number filter, and *(ii)* a customer-to-provider (c2p) filter.

Spurious number filter. This filter mitigates the presence of *spurious numbers* (numeric expressions that are not ASNs). The feature extraction rules lack semantic context to distinguish between *spurious numbers* and actual ASNs which could potentially lead to include numeric expressions that are not ASNs.

Table 2. Examples of complex feature extraction rules for the notes

ASN	Input	`regex`	output
22546	*ALSO MANAGES AS10987, 16486 AND 46498.*	`ALSO MANAGES.*`	[10987, 16486, 46498]
18200	*ASN BEHIND 18200: 2198, 17480, 45345, 45461, 56055, 56089.*	`ASN BEHIND.*`	[18200, 2198, 17480, 45345, 45461, 56055, 56089]
19750	*CRITEO ALSO MANAGES THE FOLLOWING ASNS: 44788, 53031, 55569*	`THE FOLLOWING ASNS.* ALSO MANAGES.*`	[8613, 31672]
5413	*MERGING 8613 MERGING 31672*	`MERGING .*`	[44788, 53031, 55569]
62982	*OTHER ASN'S WE CONTROL 62195, 133188, 133366*	`WE CONTROL .*`	[62195, 133188, 133366]
28263	*WE ADMINISTERED ASN 28263, 262272, 53126 AND 265079.*	`ADMINISTERED ASN .*`	[28669, 28263, 262272, 53126, 265079]
24093	*THIS ASN IS BEHIND 38195*	`IS BEHIND .*`	[38195]
7303	*OTHER ASN UNDER 7303 ARE 10481 AND 10318.*	`ASN UNDER .*`	[7303, 10481, 10318]

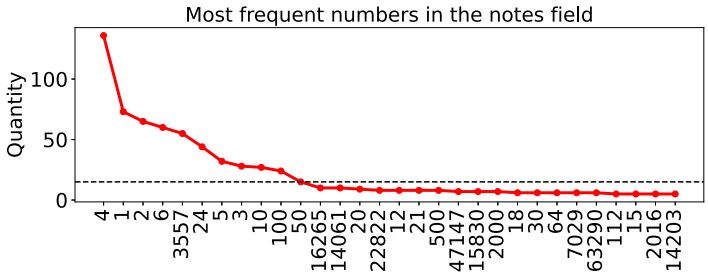


Fig. 3. Prevalence of numerical expression across the notes.

Figure 3 shows the most prevalent numeric expression across all notes of the snapshot of October 1, 2020. The most prevalent numeric expressions were extracted from notes describing protocol versions (4 and 6), maximum prefixes accepted/announced (50 or 100), and popular subnet masks (21, 22, 24 and 30 for IPv4 and 48, 64 and 80 for IPv6). The spurious-number filter includes the most prevalent number expression appearing in at least 15 notes where a knee is observed in Figure 3.

This filter also drops numbers that range between 1970 to 2020 since these numbers tend to refer to dates such as merging dates, last update, *etc.*. (*e.g.,* number of prefixes, phone numbers, addresses, years, *etc.*).

Table 3. Example of the p2c filter to filter out notes containing ASNs not related to the to network entry.

net entry	notes	inferred clustered	# c2p	decision
AS396356	*(...). Maxihost owns a Tier 3 compliant Datacenter in Sao Paulo, where its headquarter is located. We connect directly with the following ISPs, Algar (AS16735), Sparkle (AS6762), GTT (AS3257) (...)*	**396356**, 16735 (provider), 6762 (provider), AS3223 (peer), 3257 (provider), 174 (provider)	4	**drop (✗)**
AS7303	*Telecom Argentina is the major broadband and mobile provider in Argentina, with more than 4.1 MM broadband subscribers, 4 MM fixed lines and 20 MM mobile lines. Other ASN under 7303 are 10481 and 10318.*	**7303**, 10481 (provider), 10318	1	**keep (✓)**

We release our code[6] to allow users to make changes in these rules such as adding and removing them if they consider it necessary.

customer-to-provider filter. We use AS relationships to remove ASNs that are not part of the same organization. The aggressive approach could potentially group together ASNs that do not belong to the same organization but both being present in the same note. In development stage of the project, we found networks that use their notes to describe their upstream connectivity rather than listing other networks of the same organization. We then develop a stage to filter out clusters based on customer-to-provider (c2p) relationships. In our implementation, users can specify the maximum c2p relationships allowed between ASes in the same cluster or skip this stage. In cases where this filter is applied, our PDB-based inference methodology returns a file containing the list of discarded clusters. Users manually verify these cases (§4.3) and decide to either include or exclude them from the final inference.

Table 3 show this rule in action in an example in which only one c2p relationship is allowed for two different notes. In this case, the inferred cluster for Maxihost (AS396356) is dropped because this network has more c2p relationships that the maximum allowed in this example. Indeed, as the example shows, Maxihost (AS396356) is describing its upstream connectivity. On the other hand, the cluster inferred from Telecom Argentina (AS7303) meets the criteria used for this example (only one c2p allowed) and it is then preserved.

Sibling relationships generate anomalies in the inference of AS relationships [51,16,40,20]. These anomalies challenge to distinguish customer-provider relationships is between two independent companies or two companies belonging to the same conglomerate. Given that text fields can indistinctly siblings or

[6] *as2org+* can be found at: https://github.com/NU-AquaLab/as2orgplus

upstreams, we leave for human inspection those clusters containing customer-to-provider relationships across members. This inspection stage is going to assess whether discarded clusters containing customer-to-provider relationships are under the same management. In the worst case scenario, this filter would discard all sibling inferences, reducing PDB-based inference capabilities through these features to zero but in any case will infer any new cluster for AS relationships data. In any case, this filter is going to use BGP-derived data to expand the sibling inferences. In Section 6.5, we evaluate this filter with different threshold values.

4.3 Manual inspection

To conclude the data extraction process, the framework includes a last stage for human inspection to manually remove errors that were not filtered out in the previous automatic stages. This stage also allows users to apply their own judgment to filter out clusters generated by correctly extracting data, though from entries with mistakes (*e.g.,* typos).

The lack of authentication of the information given in text fields could be another source of erroneous inferences that requires human inspection. For example, we found that for a short period of time (from 2019 to 2020) a Bangladeshi provider called Brother Online (AS135131) was using its aka field to report "AS32934" (Meta's principal peering network) making our PDB-based inferencing method to group both networks together (see Appendix C). We are unaware whether this was an unintended or malicious event, though, this event highlights the sensitivity of *as2org+* to imprecise or unauthenticated data provided in text fields as well as the need of human inspection to rule out these cases.

4.4 Data consolidation

After extracting and cleaning the embedded data, the framework groups together partially overlapping clusters scattered across multiple records, fields and data sources. There are some cases in which sibling information is scattered in the same field (*e.g.,* notes) across multiple records rather than being centralized. This is illustrated in Listing 1.2 where both networks report the same parent network but none of them reference each other. Another popular case is to find sibling information scattered across multiple fields (*e.g.,* notes and org_id). This stage concludes combining clusters in our PDB-based approach with clusters in the AS2Org dataset [8] to create a dataset that we call *as2org+* .

```
# StarHub AS10091
{'asn': 10091,
 'notes': 'Please refer to as4657 PDb for Contact & Peering
     Info. Thanks.',}
# StarHub AS38861
{'asn': 38861,
```

```
6  'notes': 'Please refer to as4657 PDb for Contact & Peering
       Info. Thanks.',}
```

Listing 1.2. Example of partially overlapping clusters

5 Effectiveness of cluster extraction methods

We evaluate whether our PDB-based inference framework effectively extracts all siblings' ASNs embedded in PDB records. We focus inferences generated by using `notes` and `aka` fields and exclude `org_id` from this examination since we do not rely on heuristics to extract sibling information.

Table 4. Effectiveness of the extraction methods given by True Positive (tp), False Positive (fp), False Negative (fn), True Negative (tn), Accuracy (**A**), Precision (**P**) and Recall (**R**) values.

		Predicted notes		Predicted aka	
		Positive	Negative	Positive	Negative
Actual	Positive	446	10	230	0
	Negative	16	740	4	563
		A: 0.98 P: 0.97 R: 0.98		**A: 0.99 P: 0.98 R: 1.0**	

We manually evaluate the effectiveness of the extraction methods by analyzing whether these methods successfully extracted embedded sibling ASNs in text fields. We do not evaluate the correctness of the reported data since we lack ground truth. We consider a cluster that extracts all sibling information embedded in the fields a True Positive (tp), a cluster that contains numbers that do not correspond to ASNs (spurious numbers, prefixes, numbers in URLs, *etc.*) a False Positive (fp), a cluster that misses at least one ASN present in their corresponding text fields a False Negative (fn) and a text field that contains numeric expressions with no embedded siblings a True Negative (tn).

Table 4 shows True Positive (tp), False Positive (fp), False Negative (fn), True Negative (tn), accuracy (**A**), precision (**P**) and recall (**R**) values for the output of our inference method (using c2p threshold = 0 and without reintroducing clusters) using the snapshot of September 1, 2022. According to the results of our evaluation, our PDB-based inference framework successfully extracts embedded ASNs in text fields with values of accuracy, precision and recall of 0.98, 0.97 and 0.98 and 0.99, 0.98 and 1.0, for notes and aka, respectively.

As a result of this evaluation, we identified a list of challenges that the manual inspection stage faces to distinguish numbers corresponding to siblings. Table 5 shows some prominent examples that we gathered during this process, such as the presence of Best Current Practice (BCP) numbers, networks reporting

partnership, hosting third-party resources, adoption of cloud services, among others.

Table 5. Examples generating challenges to the manual inspection stage to assess whether some networks are under the same management.

Challenge	ASN	Example
BCP	34428	*we use filtering according to BCP38*
Partnerships	19281	*We typically partner with PCH (ASN42) in IX locations*
Third-party resources	393424	*This is the TorIX Services network which includes a (...),* **AS112 node**
Cloud-hosted services	27471	*The WSS service has migrated to Google Cloud (...) peer with Google AS15169*
Numbers in URLs	50618	*https://as29075.peeringdb.com* (✓)
	397102	*https://peeringdb.com/net/200* (✗)

6 Evaluating a PDB-based inferencing

We exhaustively evaluate the contribution of different components and stages of the PDB-based inferencing approach to the sibling inference problem. We evaluate the contribution of different features (§6.1), the aggressive approach (§6.2), simple and complex rules (§6.3) and the data consolidation stage (§6.6). We also investigate the prevalence of using text fields to report unregistered siblings (§6.4) and the impact of the c2p filter (§6.5).

For this evaluation, we run a longitudinal analysis using PDB snapshots from five different years (Sept. 3, 2018, Sept. 7, 2019, Sept. 1, 2020, Sept. 1, 2021, Sept. 1, 2022). To complete the framework setup, we include CAIDA's AS relationship files of each corresponding month and configure the c2p threshold = 0 (unless a different configuration is mentioned).

6.1 Unique contribution of features

We focus on the contribution of each feature to cluster inferences to investigate whether operators are more inclined to report *siblings* in certain fields. We evaluate the number of clusters inferred by each feature to the cluster inferences using snapshots of five different years.

We run our approach to evaluate the contribution of each feature (`notes`, `aka` and `org_id`) to sibling inferencing. Table 6 shows the number of clusters (and non-atomic clusters, *i.e.,* having more than 1 ASN, in parenthesis) obtained by each feature in different snapshots collected in the past five years. We observe that `org_id` provides more clusters than any other source, two orders of magnitude more when it is compared to results obtained by `aka` and `notes`. Narrowing

	aka		notes		org	
	$\overline{AC}$	#	$\overline{AC}$	#	$\overline{AC}$	#
'18	39	128	95	229	585	12264
'19	44	160	145	289	796	14962
'20	45	188	161	338	988	18115
'21	44	208	186	400	1171	20704
'22	48	234	214	472	1384	23191

Table 6. Non-atomic ($\overline{AC}$) and total number (#) of clusters inferred per feature.

	notes		aka		org	
	aka	org	notes	org	notes	aka
'18	2 (95)	33 (95)	3 (39)	8 (39)	30 (585)	7 (585)
'19	3 (145)	65 (145)	3 (44)	10 (44)	48 (796)	7 (796)
'20	2 (161)	75 (161)	3 (45)	12 (45)	57 (988)	8 (988)
'21	3 (186)	88 (186)	4 (44)	12 (44)	65 (1171)	8 (1171)
'22	2 (214)	106 (214)	3 (48)	11 (48)	76 (1384)	7 (1384)

Table 7. Full overlap between clusters inferred by two given pairs of features. Numbers in brackets show the total number of clusters found per each feature.

our focus to *non-atomic* clusters, `org_id` still leads, however, `aka` and `notes` now contributes 4.79% and 16.42% on average compared to `org_id` during this period. We expect a more prevalent use of `org_id` to report networks under the same management since this is a native (and compulsory) field The results also suggest that `aka` and `notes` are used to communicate relationships that are not captured by the `org_id`.

We further investigate partial overlaps between non-atomic clusters inferred using different features. We specifically look for cases where a cluster inferred by a field (*e.g.,* notes) is fully contained in a cluster inferred by another field (*e.g.,* org_id). By meeting this condition, the former field would provide no contribution since that information is available in the latter field. Table 7 shows the number of clusters inferred by each feature that are fully contained in clusters inferred by another feature. We observe that clusters inferred using `notes` and `aka` fields are rarely contained in each other. This is notably different when we compute the overlap between `notes` and `aka` with `org_id` where up to half of those clusters are contained in the `org_id`. We suspect that in these overlaps attempt to make sibling information available in text format at a glimpse. In any case, the low fractions in these overlaps suggests that each feature provides a unique contribution that is not visible by any other way.

We investigated the lack of partial overlap between text fields and the `org_id` and found that this mostly occurs after mergers and acquisitions. We suspect that this common practice allows operators to quickly communicate mergers and acquisitions rather than migrating networks to a different PDB organization. We also believe that the visibility of text fields may be more effective to inform these changes to other operators.

6.2 The aggressive approach

Next, we use the 5-year dataset to examine the aggressive inference approach to evaluate the contribution of the `aka` and `notes` fields to the sibling inference.

The contribution of the *aggressive approach* depends on the use of `aka` and `notes` fields to report sibling relationships. Given that these fields are occasion-

Table 8. Effectiveness of the aggressive approach as a function of the number of records containing data and numeric expressions.

field	snapshot	# records all	non-empty ($\bar{e}$) ($\bar{e}/all$)	w num. chars (n) (n/all)	# ASN (ASN/n)
notes	2018	13406	2034 (0.15)	1090 (0.08)	386 (0.35)
	2019	16485	2287 (0.14)	1243 (0.08)	503 (0.40)
	2020	19966	2669 (0.13)	1437 (0.07)	632 (0.44)
	2021	22892	2987 (0.13)	1622 (0.07)	743 (0.46)
	2022	25767	3326 (0.13)	1812 (0.07)	873 (0.48)
aka	2018	13406	6349 (0.47)	435 (0.03)	188 (0.43)
	2019	16485	8440 (0.51)	560 (0.03)	231 (0.41)
	2020	19966	10594 (0.53)	670 (0.03)	260 (0.39)
	2021	22892	12276 (0.54)	775 (0.03)	281 (0.36)
	2022	25767	13880 (0.54)	864 (0.03)	316 (0.37)

ally used, we examine the prevalence of records with non-empty `aka` and `notes` fields. Towards the goal of extracting siblings from these fields, we investigate the prevalence of `aka` and `notes` containing numeric expressions. We then use this information to compute the average number of ASNs extracted per record containing numeric expressions

Table 8 shows the number of `aka` and `notes` fields with non-empty records ($\bar{e}$), those containing numeric expressions (n) and the number of ASNs extracted for a 5-year period. Overall, `notes` are rarely used — only a fraction from 0.15 to 0.13 contains data — and `aka` (0.47 to 0.54) is more commonly used, however, both rarely contain numeric expressions (fractions oscillate around 0.08 and 0.03 respectively). Interestingly, the ratio between fields containing numeric expressions and the total number of ASNs extracted is between 0.3 and 0.5, showing that on average fields with numeric expressions provide 0.3 to 0.5 ASNs per field. We also observe that the fraction of non-empty records, those containing numeric expressions, are stable over time while the number of ASNs embedded in notes augmented in the same period. This growth suggests that notes are being more frequently used to report other ASes under the same management.

6.3 Simple rules, complex rules and both combined

Given the prevalence of siblings embedded in `notes` containing numeric expressions, we continue our evaluation looking at the contribution of simple rules, complex rules and both combined.

For this analysis we consider that a cluster is visible for both methods *iff* both outputs contain the same elements. For example, let A, B be two inferred clusters where A and B are inferred by simple and complex rules, respectively. We consider that both methods generate the same output if $\forall a_i \in A, a_i \in B \wedge \forall b_j \in B, b_j \in A$.

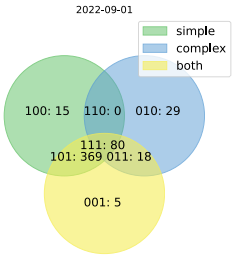


Table 9. Clusters' overlap obtained after applying simple rules, complex rules and both combined.

	PDB	notes inferences		aka inferences	
year	# ASN	#ASN	*unregistered*	#ASN	*unregistered*
2018	13406	386	36 (0.09)	188	44 (0.23)
2019	16485	503	43 (0.09)	231	46 (0.20)
2020	19966	632	43 (0.07)	260	45 (0.17)
2021	22892	743	46 (0.06)	281	43 (0.15)
2022	25767	873	51 (0.06)	316	48 (0.15)

Table 10. Number of ASNs registered in PDB, number of ASNs inferred to be in clusters in the `notes` and `aka` and the number (and fraction) of those inferred ASNs in text field that have not been registered in PDB (*unregistered*).

Table 9 shows a Venn diagram with the overlap between the clusters inferred with simple rules, complex rules and both using a snapshot collected on September 1, 2022. We observe that simple rules capture 90.4% of the clusters (464/513) while the remaining clusters are observed when complex rules are applied solo or in combination of simple rules. This is a remarkable observation since simple rules have patterns that are less prone to capture spurious numbers (we recall Table 1) and they are highly successful in extracting embedded siblings. This finding also shows that despite there being no standard format to report siblings, operators mostly use similar unsophisticated patterns. A final observation is that simple and complex rules infer some identical clusters that are not visible when both rules are combined. This behavior is due to the fact that the combination of rules can create a more rich clusters in the entire dataset and some of these new enriched clusters eventual merge and create discrepancies.

6.4 Reporting unregistered siblings

Considering that `notes` and `aka` are free text fields, we investigate the use of these fields to report siblings that are not registered in PDB.

Table 10 shows the number of ASes registered in PDB, the number of ASNs in clusters inferred from `notes` and `aka` fields and the number of those inferred ASN that have not been registered in PDB. For the 2018-2022 period, we observe that the prevalence of unregistered ASNs is more significant in `aka` than in `notes`, ranging between 0.23 and 0.15 and 0.09 and 0.06 respectively. We also note that both trends have been declining over time, though for `aka` roughly 15% of the siblings reported are not present in PDB records. We suspect that operators sometimes report unregistered ASNs in a single record to reduce management overhead associated with registration and maintenance of multiple records. Despite being convenient, reporting ASNs that are not present in PDB lacks authentication and it is unclear whether these ASNs are in fact all under the same management.

6.5 Removing upstream providers

We now shift our attention to the c2p filter to investigate the impact of different c2p threshold values. In this analysis, we evaluate the trade-off between discarding false positive inferences (*i.e.,* clustering ASNs from different organizations) and discarding correctly inferred clusters (*i.e.,* an inferred c2p relationship between ASes of the same organization).

Table 11. Impact of the c2p filter on the sibling inferences as the number of filtered clusters with different threshold values. We use three posible outcomes, *(i)* positive (ASNs were not under the same management), *(ii)* negative (ASNs were under the same management) and *(iii)* neutral (ASNs were under the same management but the same information is available through the `org_id`. Numbers in parenthesis correspond to the fraction of clusters in that category of a c2p threshold value.

	c2p threshold values					
category	0	1	2	3	4	5
Positive	10 (0.04)	3 (0.08)	3 (0.14)	2 (0.15)	1 (0.09)	2 (0.29)
Neutral	125 (0.52)	11 (0.30)	5 (0.24)	2 (0.15)	1 (0.09)	1 (0.14)
Negative	104 (0.44)	23 (0.62)	13 (0.62)	9 (0.69)	9 (0.82)	4 (0.57)

We recall that the c2p filter discards clusters (before the data consolidation stage) when the number of c2p relationships across members exceeds the threshold value (§4.2). For the evaluation, we apply five different threshold values (0-5) to the snapshot of September 1, 2022. We conduct human inspection to assess whether the cluster was successfully removed based on the text provided in the notes. Table 11 shows the results for this human inspection where filtered clusters are categorized into three types: *(i)* positive (ASNs were not under the same management), *(ii)* negative (ASNs were under the same management) and *(iii)* neutral (ASNs were under the same management but the same information is available through the `org_id`). The results show that filtered out clusters are mostly legit siblings and a small fraction of them contain networks reporting their upstream connectivity. The overlap between `notes` and `org_id` (§6.1) partially mitigates the impact of removing valid clusters.

We manually examined the filtered clusters that contain upstream providers under different managements. We found that these networks use their `notes` to list their connectivity with several large transit networks (*e.g.,* Level3-3356, Telecom Italia-6762, GTT-3257) that belong to different corporations (see an example in Appendix D). This example argues in favor of implementing the c2p filter as a mechanism to prevent our approach from clustering together high-profile networks that belong to different organizations.

To summarize this analysis, the c2p filter successfully removes false positive inferences but with the cost of also discarding clusters containing siblings. The consequence of this filter is that it introduces a human examination phase

to reintroduce the valid-but-removed clusters. We leave as future work a more refined filter that reduces the human interaction in the process.

6.6 Grouping scattered sibling information

We conclude our evaluation by looking at the effectiveness of the consolidation stage in grouping partially overlapping clusters. We investigate the number of clusters obtained after applying extraction and filtering stages that required the consolidation stage to be grouped into single clusters.

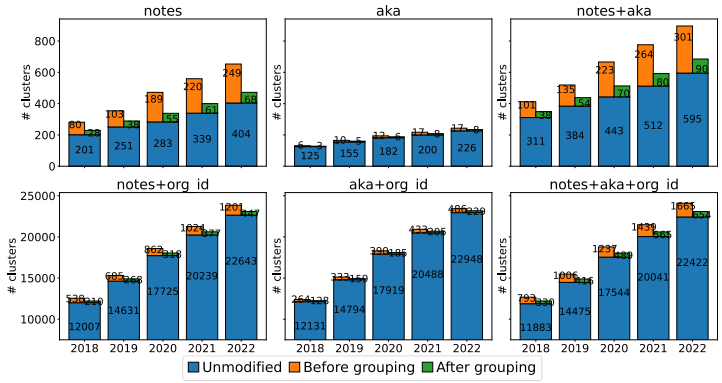


Fig. 4. Impact of the consolidation stage grouping partially overlapping clusters together.

We recall §4.3 where we describe that sibling information may be scattered across multiple PDB records. We now apply the PDB-based inferencing approach and investigate the prevalence of sibling information scattered across multiple records that creates partially overlapping clusters. Figure 4 shows the contribution of the consolidation stage counting the number of clusters before and after this stage and highlighting the number of unmodified clusters for six feature combinations in the five-year dataset. We observe that the majority of the clusters remain the same after applying this stage since the information was not scattered or they were just atomic clusters. However, for the fraction of clusters that was susceptible to be further grouped, the effectiveness of this stage is remarkable with a compression factor (number of clusters before and after the stage) between 3:1 and 4:1. This highlights the lack of uniform patterns to share sibling information as well as the prevalence of organizations without a record that aggregates all networks under control of the organization.

7 *as2org+* : Enriching the AS2Org dataset with PeeringDB

In this section we investigate the contribution of a PDB-based inferencing approach to enhance the AS2Org' AS-to-Organization mappings. We evaluate the overall contribution to the AS-level topology (§7.1), finding that the Organization-level topology is composed by 92% single-AS clusters. For the remaining 8%, responsible for delivering the majority of Internet's traffic [22], we evaluate changes in organizations of large transit networks (§7.2) and *Hypergiants* (§7.3).

7.1 Enhancing `AS2Org`

In this section we investigate the contribution of PDB-based inferencing to the existing AS-to-Organization mapping techniques as a complementary source of data. *as2org+* combines the WHOIS-based AS2Org clusters with the output of our PDB-based inferencing approach.

Table 12. Contribution of PDB-based inferencing to AS2Org datasets seen in the *as2org+* output.

				non-atomic clusters				
		# clusters(AS2Org)		# clusters		# ASes		
field	year	all	unmodif.	*as2org+*	AS2Org	*as2org+*	AS2Org	migrant ASes
notes	2018	71288	70806	5529	5729	20994	21373	1518
	2019	75223	74979	5925	5932	22498	22348	759
	2020	79126	78870	6407	6424	24529	24385	815
	2021	86565	86255	6833	6856	26052	25872	1149
	2022	90508	90144	7272	7324	27771	27580	1444
aka	2018	71288	70921	5528	5729	20917	21373	1348
	2019	75223	75122	5935	5932	22413	22348	363
	2020	79126	79022	6420	6424	24446	24385	367
	2021	86565	86454	6849	6856	25936	25872	401
	2022	90508	90402	7311	7324	27635	27580	712
org	2018	71288	70382	5526	5729	21261	21373	3168
	2019	75223	74358	5946	5932	22906	22348	2659
	2020	79126	78154	6438	6424	25002	24385	2991
	2021	86565	85474	6865	6856	26561	25872	3613
	2022	90508	89251	7338	7324	28387	27580	4150

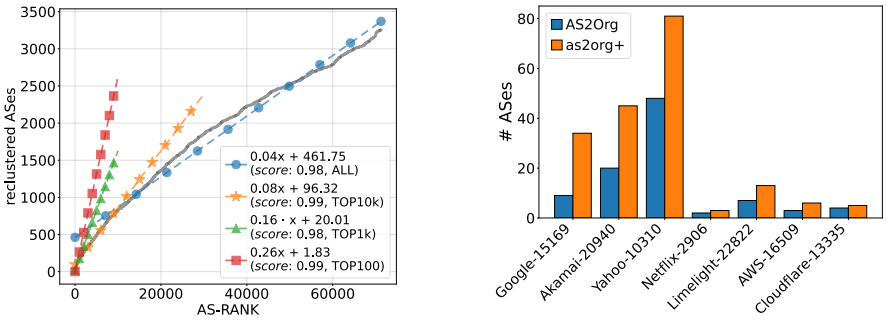
We evaluate the contribution of the PDB-based inference in *as2org+* from different perspectives. We use the AS2Org dataset as a baseline to compare it with *as2org+* to evaluate the total number of clusters that *as2org+* modifies. We then narrow the analysis and specifically examine the contribution of *as2org+* in modifying the number of non-atomic clusters. We finally contrast both datasets

from the AS-level perspective and investigate the prevalence of migrant ASes, ASes that moved into a new cluster after adding the PDB-based inference.

Table 12 shows the contribution of the PDB-based inference approach to *as2org+* when different features are used in the 5-year dataset described in §6. We observe minor modifications to the number of clusters (including non-atomic clusters), independent of the snapshot and feature used. It is worth noting that we expect to see minor changes since the Internet is mostly composed of small single-AS organizations. The number of clusters in AS2Org before and after combining it with PDB-based inferences shows minor changes too due, in part, to the impact of the consolidation stages that groups together clusters when they partially overlap. Nonetheless, the number of migrant ASes reaches 4150 (≈4% of the ASes in AS2Org database) using the `org_id` field in the 2022 snapshot. Despite these changes appearing negligible, it is important to examine what ASes and organizations are being modified by this contribution. In the next section, we explore some aspects of the network to put in perspective the impact of these changes.

7.2 Reshaping large transit organizations

In the following paragraphs we shift our attention to the contribution of *as2org+* in drawing a more complete structure of large transit organizations and hypergiants. To put that contribution in perspective, we use CAIDA's AS-RANK [7] and investigate where there is a correlation between reshaped organizations and the transit ranking of these networks.



(a) Changes in the number of *siblings* inferred as a function of CAIDA's AS-RANK.

(b) Changes in the number of *siblings* in organization operating *Hygergiants*.

Fig. 5. Contribution of *as2org+* to obtain a better representation of large transit (Fig. 5a) and content delivery (Fig 5b) organzations.

We create a N-dimensional vector $v \in \{0,1\}^N$ where N is the number of ASes in the AS-RANK and the order is given the networks' ranking. We then fill that

vector with either 0 or 1 where 1 means that that AS is now in a different cluster compared to the AS2Org dataset. Figure 5a shows the cumulative sum of the status vector v as well as linear regressions for the cumulative sum containing top100, top1k, top10k and all ASes in the AS-RANK. We observe that 3,254 out 71,258 ASes in the AS-RANK have moved into a different cluster comparing *as2org+* and AS2Org datasets. A linear equation describes, with a high accuracy (linear regression score: 0.98), the contribution of the PDB-based inference to reshape clusters. However, the curve is notably separated at the top of the ranking (seen at the beginning of the curve) indicating a different model for that portion. We then apply linear regression for top100, top1k, top10k ASes in the ranking and find that the slope coefficient increases when we narrow the selection of top-ranked networks. In numbers, the slope coefficient is 0.26, 0.16, 0.08 and 0.04 for linear regression including top100, top1k, top10k and all ASes in the AS-RANK. In other words, this means that 1 out of 4, 6, 12 and 25 has moved into a new cluster for different slices of the AS-RANK. This finding highlights that *as2org+* contribution is more prevalent across organizations operating large transit networks.

7.3 Impact in Hypergiant organizations

Last, we investigate whether our PDB-based inference approach draws a more complete representation of large content providers, also known as *Hypergiants (HGs)* [5,22,33], at an organization level. We study the contribution of PDB-based inference to the 15 most prominent HGs[7] identified by recent works on that space [5,11,10].

Figure 5b shows the 7 HGs organizations that have changed when *as2org+* data is compared to AS2Org. The contribution of the PDB-based inference approach to the representation of these HGs is not homogeneous; Yahoo!, Akamai, Google, organizations have grown in 43, 25 and 25 ASNs, respectively, while Limelight, Amazon, Netflix and Cloudflare 6, 3, 1 and 1, respectively. This new organization-level representation groups together different Google's business units (*e.g.,* Google's AS15169, Google Fiber (AS16591) and Google Cloud Services (AS396982)), Akamai's subsidiaries (*e.g.,* AS20940, Prolexic-32787 and Linode-63949) and Amazon's networks (AS16509 and AS14618). This shows that *as2org+* contributes to draw a more complete representation of the organizations serving large fractions of Internet's traffic.

8 Related Work

Despite the popularity of both WHOIS and PeeringDB datasets, to the best of our knowledge, there is no prior work that has combined both datasets to address the AS-to-Organization mapping problem.

[7] The list is composed of Apple-AS714, Amazon-AS16509, Facebook-AS32934, Google-AS15169, Akamai-AS20940, Yahoo!-AS10310, Hurricane Electric-AS6939, OVH-AS16276, LimeLight-AS22822, Microsoft-AS8075, Twitter-AS13414, Twitch-AS46489, Cloudflare-AS13335 and Edgecast-AS15133

Our work builds on the seminar work by Cai et al. [57] which created an automated methodology using WHOIS records to generate AS-to-organizations mappings, and Hyun *et al.* [29] which discusses of common practices in the use of multiple ASes for a single organization and introduces the idea of using WHOIS records to identify ASes under the same administration.

The WHOIS data received notorious attention given that this database offers information that is not embedded in network protocols interactions. To enable characterizations of the .com WHOIS data, Liu *et al.* [36] proposed parse and structure WHOIS query responses using a conditional random field model. For a different purpose, Livadariu *et al.* [37] examined WHOIS records to contrast the results of IP geolocation services finding partial overlaps in geolocation and delegated country fields.

A number of research efforts relied on PeeringDB as a source of topological data. Lodhi *et al.* [38] investigated the accuracy and representativeness of PDB records finding strong correlations between address space, traffic volume and geographic footprint in these records and other sources of network data. Bottger et al. [5] used several network features publicly reported in PeeringDB to identify the most prominent CDNs (Hypergiants). Other research efforts relied on PeeringDB's AS-to-facilities lists to detect ASes footprint and facilities outages [24,23]. In a recent work, Carisimo *et al.* [9] leveraged PeeringDB data to identify ASNs belonging to the same organization in the context of state-owned Internet Operators.

9 Conclusions and future directions

We presented $as2org+$, a new framework that leverages *self-reported* information available on PeeringDB to boost the state-of-the-art WHOIS-based methodologies, arguing that a collaborative operator-oriented database could bring a complementary perspective to the information available in WHOIS records. We conducted an in-depth study of the common practices used in PDB to report *ASes under the same management*. We apply this knowledge to design the sibling extraction rules that are at the core of the $as2org+$ framework. We evaluated the contribution of this new approach and used it to carry out a preliminary analysis showing it helps yield a better representation at the Organization level of large transit networks, multinational conglomerates and merger and acquisitions.

This work suggests several promising directions for future work including the use of ML and NLP tools. These learning approaches could leverage the semantic context of the data to refine our extraction process. These techniques could be also applied to better represent complex organizations (*e.g.,* China Telecom) with multiple registration IDs in WHOIS but minimally present on PDB.

10 Acknowledgements

This work was partly funded by the research grant CNS-2107392.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. nature **406**(6794), 378–382 (2000)
2. ARIN: Rdap: Whois for the modern world. https://www.arin.net/blog/2016/05/26/rdap-whois-for-the-modern-world/ (2016)
3. ARIN: Organization identifiers (org ids). https://www.arin.net/resources/guide/account/records/org/ (2022)
4. ARIN: Organizations holding legacy resources. https://www.arin.net/resources/guide/legacy/ (2022)
5. Böttger, T., Cuadrado, F., Uhlig, S.: Looking for hypergiants in PeeringDB. ACM SIGCOMM Computer Communication Review **48**(3), 13–19 (2018)
6. CABASE: Instructivo peeringdb. https://www.cabase.org.ar/wordpress/wp-content/uploads/2014/09/Alta-en-peeringDB.doc (2022)
7. CAIDA: As rank. https://catalog.caida.org/details/software/asrank_api, accessed: 2022-1-17
8. CAIDA: Mapping autonomous systems to organizations: Caida's inference methodology. https://www.caida.org/archive/as2org/ (2022)
9. Carisimo, E., Gamero-Garrido, A., Snoeren, A.C., Dainotti, A.: Identifying ases of state-owned internet operators. In: Proc. of IMC (2021)
10. Carisimo, E., Selmo, C., Alvarez-Hamelin, J.I., Dhamdhere, A.: Studying the evolution of content providers in the Internet core. In: Proc. of TMA. IEEE (2018)
11. Carisimo, E., Selmo, C., Alvarez-Hamelin, J.I., Dhamdhere, A.: Studying the evolution of content providers in IPv4 and IPv6 internet cores. The International Journal for the Computer and Telecommunications Industry **145**, 54–65 (2019)
12. CenturyLink: Centurylink completes acquisition of Level 3. https://news.lumen.com/2017-11-01-CenturyLink-completes-acquisition-of-Level-3 (2017)
13. Cho, S., Fontugne, R., Cho, K., Dainotti, A., Gill, P.: Bgp hijacking classification. In: Proc. of TMA. pp. 25–32 (2019). https://doi.org/10.23919/TMA.2019.8784511
14. Company, T.: Telia company's divestment of Telia carrier completed. https://www.teliacompany.com/en/news/press-releases/2021/6/telia-companys-divestment-of-telia-carrier-completed/ (2021)
15. Dhamdhere, A., Clark, D.D., Gamero-Garrido, A., Luckie, M., Mok, R.K.P., Akiwate, G., Gogia, K., Bajpai, V., Snoeren, A.C., Claffy, K.: Inferring persistent interdomain congestion. In: Proc. of ACM SIGCOMM. p. 115. SIGCOMM '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3230543.3230549, https://doi.org/10.1145/3230543.3230549
16. Dimitropoulos, X., Krioukov, D., Fomenkov, M., Huffaker, B., Hyun, Y., Claffy, K., Riley, G.: As relationships: Inference and validation. ACM SIGCOMM Computer Communication Review **37**(1), 29–40 (2007)
17. Dolev, D., Jamin, S., Mokryn, O.O., Shavitt, Y.: Internet resiliency to attacks and failures under bgp policy routing. Computer Networks **50**(16), 3183–3196 (2006)
18. Facebook: Peering: Technical requirements. https://www.facebook.com/peering (2022)
19. Feamster, N., Winick, J., Rexford, J.: A model of bgp routing for network engineering. In: Proc. of ACM SIGMETRICS. p. 331342. SIGMETRICS '04/Performance '04, Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1005686.1005726, https://doi.org/10.1145/1005686.1005726

20. Gao, L.: On inferring autonomous system relationships in the Internet. IEEE/ACM Transactions on Networking **9**(6), 733–745 (2001)
21. Gao, L., Rexford, J.: Stable internet routing without global coordination. Proc. of ACM SIGMETRICS **28**(1), 307317 (jun 2000). https://doi.org/10.1145/345063.339426, https://doi.org/10.1145/345063.339426
22. Gigis, P., Calder, M., Manassakis, L., Nomikos, G., Kotronis, V., Dimitropoulos, X., Katz-Bassett, E., Smaragdakis, G.: Seven years in the life of hypergiants' offnets. In: Proc. of ACM SIGCOMM (2021)
23. Giotsas, V., Dietzel, C., Smaragdakis, G., Feldmann, A., Berger, A., Aben, E.: Detecting peering infrastructure outages in the wild. In: Proc. of ACM SIGCOMM (2017)
24. Giotsas, V., Smaragdakis, G., Huffaker, B., Luckie, M., Claffy, K.: Mapping peering interconnections to a facility. In: Proc. of CoNEXT (2015)
25. Google: Prerequisites to peer with Google. https://peering.google.com/#/options/peering (2022)
26. Greenlees, D., Arnold, W.: Asia scrambles to restore communications after quake - business - international herald tribune. https://www.nytimes.com/2006/12/28/business/worldbusiness/28iht-connect.4042439.html (2006)
27. Holz, R., Hiller, J., Amann, J., Razaghpanah, A., Jost, T., Vallina-Rodriguez, N., Hohlfeld, O.: Tracking the deployment of tls 1.3 on the web: A story of experimentation and centralization. ACM SIGCOMM Computer Communication Review **50**(3), 315 (jul 2020). https://doi.org/10.1145/3411740.3411742, https://doi.org/10.1145/3411740.3411742
28. Huston, G.: The death of transit and the future internet. In: ITU Workshop on Network. vol. 2030 (2018)
29. Hyun, Y., Broido, A., claffy, k.: Traceroute and BGP AS path incongruities. Tech. rep., Cooperative Association for Internet Data Analysis (CAIDA) (2003-03)
30. ICANN: Registration data access protocol (RDAP). https://www.icann.org/rdap (2022)
31. Jin, Y., Scot, C., Dhamdhere, A., Giotsas, V., Krishnamurthy, A., Shenker, S.: Stable and practical AS relationship inference with ProbLink. In: Proc. of USENIX NSDI (2019)
32. Kashaf, A., Sekar, V., Agarwal, Y.: Analyzing third party service dependencies in modern web services: Have we learned from the mirai-dyn incident? In: Proc. of IMC. p. 634647. IMC '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3419394.3423664, https://doi.org/10.1145/3419394.3423664
33. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. ACM SIGCOMM Computer Communication Review **40**(4), 75–86 (2010)
34. Laskowski, P., Chuang, J.: Network monitors and contracting systems: competition and innovation. Proc. of ACM SIGCOMM **36**(4), 183–194 (2006)
35. Liu, E., Akiwate, G., Jonker, M., Mirian, A., Savage, S., Voelker, G.M.: Who's got your mail? characterizing mail service provider usage. In: Proc. of IMC. p. 122136. IMC '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3487552.3487820, https://doi.org/10.1145/3487552.3487820
36. Liu, S., Ian, Foster, Savage, S., Voelker, G., Saul, L.: Who is. com? learning to parse WHOIS records. In: Proc. of IMC. pp. 369–380 (2015)

37. Livadariu, I., Dreibholz, T., Al-Selwi, A.S., Bryhni, H., Lysne, O., Bjørnstad, S., Elmokashfi, A.: On the accuracy of country-level IP geolocation. In: Applied Networking Research Workshop. pp. 67–73 (2020)
38. Lodhi, A., Larson, N., Dhamdhere, A., Dovrolis, C., kc Claffy: Using peeringDB to understand the peering ecosystem. ACM SIGCOMM Computer Communication Review **44**(2), 20–27 (2014)
39. Lodhi, A.H.: The economics of Internet peering interconnections. Ph.D. thesis, Georgia Institute of Technology (2014)
40. Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., Claffy, K.: As relationships, customer cones, and validation. In: Proc. of IMC (2013)
41. Microsoft: Prerequisites to set up peering with Microsoft. https://docs.microsoft.com/en-us/azure/internet-peering/prerequisites (2022)
42. Moura, G.C.M., Castro, S., Hardaker, W., Wullink, M., Hesselman, C.: Clouding up the internet: How centralized is dns traffic becoming? In: Proc. of IMC. p. 4249. IMC '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3419394.3423625, https://doi.org/10.1145/3419394.3423625
43. NCC, R.: Youtube hijacking: A ripe ncc ris case study. https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study (2008)
44. Nemmi, E.N., Sassi, F., La Morgia, M., Testart, C., Mei, A., Dainotti, A.: The parallel lives of autonomous systems: ASN allocations vs. BGP. In: Proc. of IMC. pp. 593–611 (2021)
45. NIC.br: Collaboration between NIC.br and PeeringDB is helping improve Internet traffic exchange in Brazil. https://nic.br/noticia/releases/collaboration-between-nic-br-and-peeringdb-is-helping-improve-internet-traffic-exchange-in-brazil/ (2020)
46. Nobile, L., Morris, T.: Status and solutions for WHOIS data accuracy. https://archive.nanog.org/sites/default/files/3_Nobile_Whois_Data_Accuracy.pdf (2022)
47. PeeringDB: Approving network (net) objects. https://docs.peeringdb.com/committee/admin/approval-guidelines/#approving-network-net-objects (2022)
48. PeeringDB: Peeringdb api documentation. https://www.peeringdb.com/apidocs/#operation/create%20net (2022)
49. Quoitin, B., Pelsser, C., Bonaventure, O., Uhlig, S.: A performance evaluation of bgp-based traffic engineering. International Journal of Network Management **15**(3), 177–191 (2005). https://doi.org/https://doi.org/10.1002/nem.559, https://onlinelibrary.wiley.com/doi/abs/10.1002/nem.559
50. Spring, N., Mahajan, R., Anderson, T.: The causes of path inflation. In: Proc. of ACM SIGCOMM. p. 113124. SIGCOMM '03, Association for Computing Machinery, New York, NY, USA (2003). https://doi.org/10.1145/863955.863970, https://doi.org/10.1145/863955.863970
51. Subramanian, L., Agarwal, S., Rexford, J., Katz, R.H.: Characterizing the Internet hierarchy from multiple vantage points. In: Proc. of IEEE INFOCOM (2002)
52. Testart, C., Richter, P., King, A., Dainotti, A., Clark, D.: Profiling bgp serial hijackers: Capturing persistent misbehavior in the global routing table. In: Proc. of IMC. p. 420434. IMC '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3355369.3355581, https://doi.org/10.1145/3355369.3355581
53. Testart, C., Richter, P., King, A., Dainotti, A., Clark, D.: To filter or not to filter: Measuring the benefits of registering in the rpki today. In: Sperotto, A., Dainotti,

A., Stiller, B. (eds.) Proc. of PAM. pp. 71–87. Springer International Publishing, Cham (2020)
54. USA, T.M.: Tmobile completes merger with sprint to create the new tmobile. https://www.t-mobile.com/news/un-carrier/t-mobile-sprint-one-company (2020)
55. Wire, B.: Gtt acquires nlayer communications, inc. https://www.businesswire.com/news/home/20120501005550/en/GTT-Acquires-nLayer-Communications-Inc. (2012)
56. Wu, J., Zhang, Y., Mao, Z.M., Shin, K.G.: Internet routing resilience to failures: analysis and implications. In: Proc. of CoNEXT. pp. 1–12 (2007)
57. Xue, X., Heidemann, J., Krishnamurthy, B., Willinger, W.: Towards an AS-to-organization map. In: Proc. of IMC (2010)

A Example of a aka reporting siblings

Listing 1.3 shows the `net` entry of Telecom Argentina's AS7303 as examples of the use of the field `aka` to report *siblings*.

```
{"meta":
 {"generated": },
  "data": [
   {
    "asn": 7303,
    "website":"",
    "notes": "Telecom Argentina is the major broadband and mobile provider in
         Argentina, with more than 4.1 MM broadband subscribers, 4 MM fixed
        lines and 20 MM mobile lines. Other ASN under 7303 are 10481 and
        10318.",
    "org_id": 1419,
    "policy_url": "",
    "aka": "FiberCorp, Cablevision (other ASN: 10481 and 10318)",
}}}
```

Listing 1.3. Example of the `net` entry for AS7303 in the PDB snapshot of October 1, 2020.

B Examples of the aka feature extraction

Table 13 shows examples in which operators use the field `aka` to report *siblings* and the results obtained after applying the extraction rules.

ASN	Input	`regex`	output
25751	*Mediaplex, Commission Junction, FastClick, Dotomi, ValueClick, SET.tv, 41041, 26762, 19834*	`\d{4,8}`	[41041, 26762, 19834]
24130	*9722 18398 23741 23745 17999 9894 (IX Services)*	`\d{4,8}`	[9722, 18398, 23741, 23745, 17999, 9894]
8100	*FKA AS29761*	`\d{4,8}`	[29761]
714	*Apple CDN AS6185*	`\d{4,8}`	[6185]

Table 13. Examples of `regex` bieng applied to extract siblings from `aka` field.

C Example of lack of trust in reported data

Listing 1.4 shows the `net` entry of the Bangladeshi provider Brothers Online (AS135131) that was mistakenly reporting Meta's AS32934 in its `aka` field.

```
{"meta":
 {"generated": },
  "data": [
   {
    "asn": 135131,
    "website":"http://www.brothersonlineisp.com",
    "notes": "",
    "org_id": 20630,
    "policy_url": "http://www.brothersonlineisp.com",
    "aka": "AS32934",
}}}
```

Listing 1.4. `net` entry of AS135131 in the PDB snapshot of October 1, 2020.

D Example of a network reporting transit connectivity

Listing 1.5 shows the `net` entry of the CacheFly (AS30081) that includes in its notes ASNs that are not under the same management.

```
{"meta":
 {"generated": },
  "data": [
   {
 'asn': 30081,
 'name': 'CacheFly',
 'notes': 'AS3257/AS7922/AS1299/AS2914/AS1221 announces best anycast route at
       '
          'all locations in addition to direct peering.\n'
          '\n'
          'Please note we only peer with local/regional carriers in each '
          'location.',
}}
```

Listing 1.5. `net` entry of 30081 in the PDB snapshot of October 1, 2020.