Automatic Inference of BGP Location Communities

BRIVALDO A. DA SILVA JR, Universidade Federal de Mato Grosso do Sul, Brazil
PAULO MOL, Universidade Federal de Minas Gerais, Brazil
OSVALDO FONSECA, Universidade Federal de Minas Gerais, Brazil
ÍTALO CUNHA, Universidade Federal de Minas Gerais, Brazil
RONALDO A. FERREIRA, Universidade Federal de Mato Grosso do Sul, Brazil
ETHAN KATZ-BASSETT, Columbia University, USA

The Border Gateway Protocol (BGP) orchestrates Internet communications between and inside Autonomous Systems. BGP's flexibility allows operators to express complex policies and deploy advanced traffic engineering systems. A key mechanism to provide this flexibility is tagging route announcements with BGP communities, which have arbitrary, operator-defined semantics, to pass information or requests from router to router. Typical uses of BGP communities include attaching metadata to route announcements, such as where a route was learned or whether it was received from a customer, and controlling route propagation, for example to steer traffic to preferred paths or blackhole DDoS traffic. However, there is no standard for specifying the semantics nor a centralized repository that catalogs the meaning of BGP communities. The lack of standards and central repositories complicates the use of communities by the operator and research communities. In this paper, we present a set of techniques to infer the semantics of BGP communities from public BGP data. Our techniques infer communities related to the entities or locations traversed by a route by correlating communities with AS paths. We also propose a set of heuristics to filter incorrect inferences introduced by misbehaving networks, sharing of BGP communities among sibling autonomous systems, and inconsistent BGP dumps. We apply our techniques to billions of routing records from public BGP collectors and make available a public database with more than 15 thousand location communities. Our comparison with manually-built databases shows our techniques provide high precision (up to 93%), better coverage (up to 81% recall), and dynamic updates, complementing operators' and researchers' abilities to reason about BGP community semantics.

CCS Concepts: • **Networks → Routing protocols**; • **Computing methodologies → Model development and analysis**.

Additional Key Words and Phrases: Border Gateway Protocol (BGP), BGP Communities, Internet Routing

ACM Reference Format:
Brivaldo A. da Silva Jr, Paulo Mol, Osvaldo Fonseca, Ítalo Cunha, Ronaldo A. Ferreira, and Ethan Katz-Bassett. 2022. Automatic Inference of BGP Location Communities. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 1, Article 3 (March 2022), 23 pages. https://doi.org/10.1145/3508023

Authors' addresses: Brivaldo A. da Silva Jr, brivaldo.junior@ufms.br, Universidade Federal de Mato Grosso do Sul, Av. Costa e Silva, Cidade Universitária S/N, Campo Grande, MS, Brazil, 79070-900; Paulo Mol, paulomol@dcc.ufmg.br, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, ICEx/DCC, Belo Horizonte, MG, Brazil, 31270-901; Osvaldo Fonseca, osvaldo.morais@dcc.ufmg.br, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, ICEx/DCC, Belo Horizonte, MG, Brazil, 31270-901; Ítalo Cunha, cunha@dcc.ufmg.br, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, ICEx/DCC, Belo Horizonte, MG, Brazil, 31270-901; Ronaldo A. Ferreira, ronaldo.ferreira@ufms.br, Universidade Federal de Mato Grosso do Sul, Av. Costa e Silva, Cidade Universitária S/N, Campo Grande, MS, Brazil, 79070-900; Ethan Katz-Bassett, ethan@ee.columbia.edu, Columbia University, 500 West 120th Street, New York, NY, USA, 10027.

2476-1249/2022/3-ART3 $15.00
https://doi.org/10.1145/3508023

1 INTRODUCTION

The Internet is composed of *Autonomous Systems* (ASes) that exchange reachability information using the Border Gateway Protocol (BGP) [46, 48], its *de facto* interdomain routing protocol. The BGP best-path selection algorithm is flexible and allows network operators to rank routes based on policies and economic agreements. However, the growing needs for reliability and performance have led to more dynamic and complex routing policies [20, 50, 56, 63], stressing BGP and exposing the limitations of a protocol last updated more than two decades ago [46].

To overcome the limitations in BGP expressiveness, network operators have increasingly relied on the optional BGP communities attribute to convey information in their route announcements. BGP communities can encode information such as the city or router where a route was learned [19, 40], the business relationship with the neighboring network the route was learned from [20, 30, 41], or requests for actions such as BGP prepending or targeted route filtering [7, 56, 64]. Informational communities facilitate identification and troubleshooting of anomalies such as path changes [19] or routing detours [40]. For example, operators can use communities to identify the points of presence (PoPs) or links traversed by a route and infer if more preferred or better performing alternate routes exist. Action communities allow the deployment of more complex traffic engineering, helping customize routing decisions at a much finer granularity than is possible by inspecting the AS path of a route announcement. For example, operators can use communities to adjust routing parameters (*e.g.,* the LocalPref or the AS path length) or prevent route propagation to specific networks or geographic regions [2, 56].

Figure 1 illustrates an example where AS V uses a location community to control route selection. The origin AS O announces prefixes to AS A at different locations L_1 and L_2, and to AS B at location L_2. AS A tags routes received at L_1 and L_2 with communities A:L1 and A:L2, respectively. AS A announces to AS V only the route it selects as the best according to its internal policies, *i.e.,* AS V receives one route from AS A with either tag A:L1 or A:L2. Suppose that AS V has a policy that dictates that routes learned from AS B should have higher priority than routes learned from AS A, *e.g.,* because B's transit costs are cheaper than A's. However, AS V may decide to use routes received from A that traverse L_1, *e.g.,* because they have better performance that justify the higher cost. To implement this policy, AS V sets LocalPref to 120 in all routes received from AS A with location community A:L1, sets LocalPref to 100 for routes learned from AS B, and sets LocalPref of other routes to 80 (including routes from A tagged with A:L2). As BGP uses LocalPref as the first criterion to decide the best route, AS V selects the high-performance route from AS A when it traverses L_1 and the cheaper route from AS B otherwise. Routes from AS A with tag A:L2 are chosen only when no route is available from B (e.g., due to failures).

Unfortunately, the BGP communities attribute is an opaque identifier and its semantics are neither standardized nor follow any universal rule. Therefore, network operators are free to decide community values and semantics. A network A may use community A:X for triggering BGP AS-path prepending, while another network B may use community B:X for a completely different purpose, *e.g.,* signal that a route was learned in New York. Some networks catalog their communities in Internet Routing Registry (IRR) databases [58] or webpages (*e.g.,* [16]), but we cannot find documentation for most communities observed in the wild (§5). Using a manually built database of documented communities from 10 Tier-1 and 5 Tier-2 ASes that publicize their communities, we were able to classify only 56.4% of these ASes' communities observed in BGP route announcements.

The lack of standardization and public databases mapping community values to their semantics hinders the manipulation of routes for traffic engineering or the development of tools that take advantage of metadata in BGP communities. Operators have to resort to ad-hoc information in IRR databases or webpages, which may be incomplete, outdated, or available only by contacting

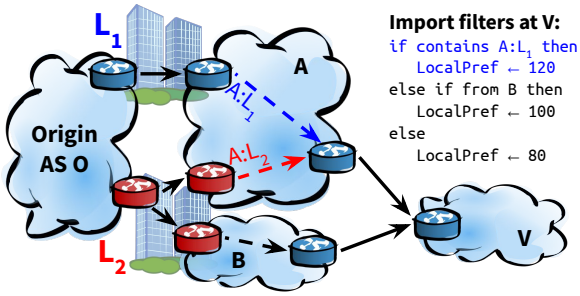


Fig. 1. Example of traffic engineering using BGP communities. AS V prefers routes from AS B, but may configure import filters to prefer routes from A when they traverse location L_1, *e.g.,* when performance through AS A and location L_1 justifies choosing the less preferred neighbor. This policy can be implemented in AS V by inspecting the location communities in AS A's route announcements.

the network operators of the particular AS. This manual process increases the effort required to integrate community information in routing decisions, degrades user quality of experience when BGP chooses suboptimal routes, and limits researchers' understanding of routing.

In this paper, we bridge this gap by developing techniques to automatically infer BGP *location communities,* defined as communities that carry metadata about the location (*e.g.,* city, country, continent, router, PoP, link, or interconnection) where a route was learned, and building a public database of BGP location communities. Location communities allow richer manipulation inside the tagging AS, but they would also be helpful to neighboring and remote ASes if their semantics were publicly available. We focus on location communities because they represent the majority of publicly-documented communities (§4) as well as a significant fraction of communities observed in route announcements (§5). Also, the flattening of the Internet hierarchy has led networks to interconnect through multiple physical links, and information about locations traversed by routes improves operators' ability to monitor policy compliance, detect unexpected behavior such as route changes, and troubleshoot anomalous behavior such as congestion. For example, operators could use a tool that correlates BGP location communities and performance to tune their route selection preferences at a finer granularity than possible with just AS paths.

A recent effort proposes a mining tool to automatically build a database of BGP community semantics by crawling information in IRR records or the support webpages of network providers [18]. The tool uses natural language processing to infer the meaning of each documented community in those data sources. While the results in [18] show that the tool achieves good precision in the extracted communities, the approach is limited *(i)* in the number of communities that it can infer, as it relies on free text descriptions provided by network operators; and *(ii)* by the data sources, which may be incomplete, outdated, or missing entirely, reducing precision of the inferred communities and coverage of communities used in the Internet.

We take a fundamentally different approach. We propose an algorithm to automatically infer location communities from public route announcements observed by BGP route collectors (*e.g.,* RouteViews [42], RIPE RIS [47], and Isolario [25]). Our key insight is to use the sequence of ASes connecting a *tagging* AS (i.e., an AS that tags routes with its location communities) to origin ASes as a reliable marker for routes crossing specific interconnection points. We use BGP route collector peers as *vantage points* from which we observe tagging ASes and correlate BGP communities with AS paths in route announcements. We also propose a set of heuristics to filter noise introduced by misbehaving networks, sharing of BGP communities among sibling autonomous systems, and

inconsistent BGP dumps. We process over two billion route announcements from three route collector projects [25, 42, 47] and infer 15,505 location communities across 1,120 ASes.

We evaluate our inference methodology using a manually built ground-truth dataset with 39,308 communities from Tier-1 and Tier-2 autonomous systems that publicize the semantics of their communities on IRR databases or webpages. Our experimental evaluation shows that our methodology yields high precision (up to 93%) and recall (up to 81%). We compare our results with CAIDA's manually-built public database of BGP communities [4] and show that our database has higher recall and similar precision, with the advantage that it can be automatically updated as new BGP communities are defined or as definitions change over time. Our code and databases of inferred and ground-truth BGP communities is available online to allow for reproducibility of our results and enhance the understanding of Internet routing by network operators and researchers [32].

1.1 Key Contributions

In this paper, we make the following key contributions:

- The design and evaluation of algorithms for automatically inferring BGP location communities from public BGP dumps, which we make available online [32]. Our evaluation shows that the algorithms achieve high precision (up to 93%) and recall (up to 81%).
- A public database with 15,505 location communities from 1,120 ASes built using our inference algorithms, as well as a manually-built ground-truth dataset identifying the semantics of BGP communities of 10 Tier-1 and 5 Tier-2 ASes.
- A characterization of the increasing use of BGP communities in the Internet. We analyze route announcements between 2017 and 2020 and show that the number of visible BGP communities increased by 51.9%, that the number of visible location communities increased by 50.0%, and that the number of ASes defining communities increased by 26.5%.

2 BACKGROUND

BGP is the interdomain routing protocol of the Internet and is used for exchanging routing information between *Autonomous Systems (ASes)—i.e.,* networks operated independently and generally by different entities—and between routers inside an autonomous system. BGP routers exchange messages that carry one or more BGP attributes. Some attributes are mandatory, such as the *AS path*, *IP prefix*, and *next hop*, and others are optional, such as *communities* and *multi-exit discriminators*. The AS path contains a sequence of *Autonomous System Numbers (ASNs)* that describes the route on the way to the *origin* network controlling and announcing the IP prefix, and the next hop contains the IP address of the next router for the announced IP prefix. The optional attributes can be transitive or non-transitive. Transitive attributes accumulate and are carried along with the route, while non-transitive attributes are processed by BGP routers at the next AS but not forwarded to upstream neighbors.

The BGP route selection algorithm uses mandatory attributes to select the preferred route to a destination IP prefix. When a router receives two BGP announcements for the same prefix, it uses a sequence of criteria to decide which route it selects. The first criterion is the *local preference* (LocalPref), which is locally defined, usually from the business relationship with the neighbor the route was received from. The Gao-Rexford model [14] defines two types of business relationships for neighboring ASes: *customer-provider* and *peer-to-peer*. A customer AS pays a provider AS for transit, *i.e.,* accessing the Internet, while peer-to-peer relationships occur when ASes have a settlement-free peering agreement where they exchange traffic free-of-charge. Autonomous systems can also have a *sibling* relationship [13]. Two autonomous systems are siblings if they are owned or operated by the same organization, share operational practices, and exchange traffic without cost or routing

restrictions. The number of sibling ASes in the Internet has grown significantly in the last few years due to acquisition or merging operations between network providers [10, 23, 41].

To implement economically favorable policies, an AS usually sets local preferences so that the BGP best-path selection algorithm prefers routes learned from customers over routes learned from peers, and prefers routes learned from peers over routes learned from providers. The selection algorithm uses other tiebreakers for routes learned from neighbors with the same relationship, such as routes with shorter AS path lengths, lower origin code, and lowest multi-exit discriminator (MED) [46]. In the Gao-Rexford model [13], the type of neighbor also determines how routes are exported. An AS exports routes learned from its customers to all neighbors, but it exports routes learned from providers and peers only to customers. Exporting routes learned from a provider or peer to other providers or peers is normally undesirable, as it would make the AS offer transit to peers and providers without monetary compensation.

The BGP optional attributes are generally used for signaling information between routers and are not used in the route selection process by default. However, they can influence the selection algorithm to implement specific policies. For example, BGP communities have been used to restrict route propagation to within a region or to influence peer selection by prepending the AS path to make the route artificially less attractive.

A route announcement can carry any number of BGP communities. Each BGP community is 32 bits long and can carry any value representing an *action* or *information* [39]. The practical convention dictates that the first 16 bits represent the ASN of the AS that defines the community's semantics, also known as *Global Administrator*, and that the last 16 bits is an arbitrary operator-defined value [39].[1] The format used in documentation or in router configurations separates the two 16-bit numbers with a colon. For example, AS3491's operators defined that the community 3491:3000 signals (and is tagged on) routes received from peers in Europe [54].

Action communities influence the BGP selection process or how routing announcements propagate. They generally signal an action that a transit provider should execute on behalf of a customer to realize some traffic engineering policy. Examples include LocalPref adjustment, BGP AS path prepending, selective advertisement, route suppression, and traffic blackholing [56].

Informational communities include metadata to a route announcement to assist operators with troubleshooting issues, refining policies, or capacity planning. Examples include tagging a route to inform that it was originated either internally or learned externally, marking the location where the route was learned, or whether the route was learned from a customer, provider, or peer. Informational communities may be used by the tagging AS itself as well as downstream ASes.

In this paper, we infer *location communities*, which are related to where a route was learned or where it goes through. Location communities can tag specific links, routers, Points of Presence (PoPs), Internet Exchange Points (IXPs), or geographical locations (*e.g.,* city, state, country, or continent). We define a *geolocation community* as one that tags a geographical location.

The use of communities has increased significantly in the past few years (§5, [56]). However, determining the semantics of each community value is a daunting task. Previous efforts have proposed standardization and better use of BGP communities to improve security [45], but operators have not fully embraced these proposals. Only a handful of community values have been standardized [29, 35, 39]. For example, 65535:666 (blackhole) signals a request to an upstream network that traffic to a destination prefix should be dropped [35], and 65535:65284 (no-advertise) signals a request to a provider that a route should not be advertised further [39]. Standardized communities

[1] In this paper, we consider only 32-bit communities [39], which use ASes with 2-byte AS numbers. BGP large communities [26] are 96-bit long and include support for 4-byte AS numbers, but their use remains incipient. Although we do not analyze large communities in this work, our techniques can be applied without modification to infer location large communities.

cover only a tiny fraction of the communities visible in route announcements. Unfortunately, no central database exists with the documentation of the existing communities. Network providers catalog their communities in ad-hoc documents or in IRR databases; and some third-party websites such as One Step aggregate this information [55]. The lack of documentation on communities and the ad-hoc nature of available documentation constrains our understanding of Internet routing.

A few existing projects—*e.g.,* University of Oregon Route Views (RV) [42], RIPE NCC Routing Information Service (RIS) [47], Isolario [25], and Packet Clearing House (PCH) [28]—collect BGP routing tables and BGP updates at several locations to help researchers and network operators better understand how BGP announcements reach different parts of the Internet. These projects deploy tens of routers that collect BGP updates from hundreds of collaborating ASes and generate datasets with millions of records every day. The datasets are openly available and valuable assets for the research and operator communities. Since the projects' routers reside in physical facilities in different parts of the world, we can use them to define *vantage points* to observe routes tagged with BGP communities. We use the term *vantage point (VP)* to refer to a router that peers with a BGP collector, and an AS might have multiple routers peering with BGP collectors at different locations.

3 INFERRING BGP LOCATION COMMUNITIES

We infer location communities based on the fact that ASes peer at a finite set of locations and enforce dynamic but deterministic routing policies [1, 14, 20, 30, 41]. We first provide an overview of the key ideas in our inference algorithm using the example in Figure 2 (§3.1) and then present our algorithm formally (§3.2).

3.1 Overview

Consider a target AS T that tags received routes with location communities (see Figure 2). If AS T and AS N_1 interconnect at a single location, then T will tag *all* routes received from N_1 with the location community corresponding to their single interconnection. The idea that all routes received at a specific location will have the corresponding location communities is the core of our algorithm. Unfortunately, we cannot simply infer communities that appear on all routes received from a neighbor N_1 as location communities. First, neighbor N_1 may tag all of its announcements with AS T traffic engineering communities, which would be incorrectly inferred as location communities. Second, when AS T and AS N_2 interconnect at multiple different locations (indicated by the multiple links between T and N_2 in Figure 2), then T may choose routes received from N_2 at any of these locations. Each chosen route will have a different location community corresponding on the interconnection over which it was received. No community will appear in all routes, and no location community would be inferred. It is challenging to infer the number of interconnections between two ASes [22], and so we do not want our approach to rely on that information.

We relax the requirement of a single interconnection and avoid the need for quantifying the number of interconnections between the target AS T and neighboring ASes by looking at paths that traverse multiple interconnections. Suppose that AS T and AS N_3 interconnect at multiple locations and that AS T receives routes with AS paths traversing $\langle N_3, N_4, N_5 \rangle$ (blue dashed line in Figure 2). Let $I_{T,3}$, $I_{3,4}$, and $I_{4,5}$ be the interconnections traversed by the routes. Interconnection $I_{T,3}$ is constrained by the set of interconnections between ASes T and N_3 *and* their routing policies. Here is a non-exhaustive list of such constraints:

(1) AS T might use multi-exit discriminators (MEDs) as a tie-breaker [46] and choose routes from N_3 received at a particular interconnection. For example, if N_3 prefers to receive traffic

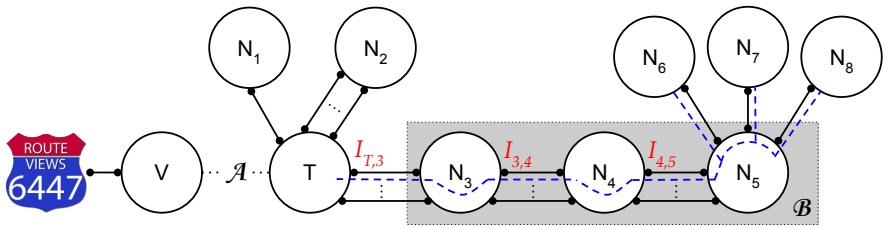


Fig. 2. Example of how long sequences of ASes between origins and a target AS T constrain the set of locations of routes received and chosen by AS T. We denote the (possibly empty) sequence of ASes between the BGP collector peer V and target AS T as $\mathcal{A}$ and the nonempty sequence of ASes constraining the locations where T may receive a BGP announcement as $\mathcal{B}$ (highlighted in gray). Solid black lines denote interconnections between ASes. In this example we assume that interconnections are at different locations, but this is not required by our algorithm.

from AS T towards $I_{3,4}$ at $I_{T,3}$, it may set lower MED values on routes exported at $I_{T,3}$, leading AS T to choose routes received at $I_{T,3}$ over routes received at other interconnections.

(2) Routers systematically choose routes from the closest (lowest IGP cost [46]) interconnection. For example, if $I_{T,3}$ is the closest interconnection to AS T's egress router towards the vantage point at V, then the egress router will choose and export routes from N_3 received at $I_{T,3}$.

(3) Routes may not be accepted by AS T or exported by AS N_3 at some interconnections, particularly when ASes T and N_3 have a complex peering relationship [20]. For example, if T and N_3 peer in Europe, but T buys transit from N_3 in the US, T will receive routes from N_3's peers and providers only in the US (*e.g.*, $I_{T,3}$).

The constraints imposed by the set of interconnections and routing policies between each pair of ASes in a route compound over consecutive AS hops. In particular, interconnection $I_{3,4}$ is constrained by the interconnections between ASes N_3 and N_4 as well as their routing policies; and similar constraints apply to $I_{4,5}$. AS T's routes exported towards vantage point V that traverse a constraining sequence of ASes (like $\langle N_3, N_4, N_5 \rangle$) will only be received by AS T at a small set of locations, possibly a single one. Looking at the problem another way, for VP V to observe routes from AS T traversing $\langle N_3, N_4, N_5 \rangle$ and received at different interconnections, then N_3 needs to receive and choose routes through $\langle N_4, N_5 \rangle$ at different interconnections, which implies N_4 receives and chooses routes from N_5 at different interconnections.

We sidestep incorrect inferences for origins that tag all their announcements with traffic engineering communities by combining observations on multiple routes from different origins. In the example, routes originated by ASes N_6, N_7, and N_8 reach AS T through the same sequence of transit ASes. The chance that *all* these origins tag their announcements with AS T traffic engineering communities is low, which allows us to correctly remove traffic engineering communities from the set of inferred location communities. In our algorithm, we require routes from a configurable number of different origin ASes to infer location communities.

3.2 Inference Algorithm

Our algorithm looks for routes from multiple origins (*e.g.*, N_6, N_7, and N_8 in Figure 2) traversing an overlapping sequence of ASes before reaching a target AS T (*e.g.*, $\langle N_3, N_4, N_5 \rangle$ in Figure 2), and infers communities from T that appear on a significant fraction of routes as location communities.

We split a route's AS path into five segments $\langle V, \mathcal{A}, T, \mathcal{B}, \mathcal{S} \rangle$, where V is the AS containing the vantage point, T is the target AS whose location communities we will infer, $\mathcal{A}$ is a possibly-empty sequence of ASes between V and T, $\mathcal{B}$ is a nonempty sequence of ASes following T, and $\mathcal{S}$ is a

Table 1. Summary of Notation.

Var	Description
V	AS hosting a BGP vantage point
T	Target AS whose location communities we are inferring
$\mathcal{A}$	Sequence of ASes between V and a target AS T
$\mathcal{B}$	Sequence of ASes after AS T constraining route propagation
$\mathcal{S}$	Suffix containing all ASes after $\mathcal{B}$ up to the origin AS
$\mathcal{R}$	Set of routes traversing a sequence of ASes
$\mathcal{R}_c$	Set of routes tagged with community c
$\mathcal{R}_T$	Set of routes traversing AS T or any of T's siblings
K_{origins}	Minimum number of distinct origins in $\mathcal{R}$ for inference
K_{prev}	Minimum fraction of routes in $\mathcal{R}$ with community for inference (prevalence)
K_{filter}	Maximum hitting set size over routes with location communities that do not traverse the community's AS or any of its siblings

nonempty suffix containing all ASes after $\mathcal{B}$ up to and including the origin AS. We consider that $\mathcal{B}$ constrains route propagation and the interconnections where AS T's chosen routes are received. AS V may be considered the target T, in which case $V = T$. An announcement needs to have an AS path with at least three ASNs to support inferences. In the cases with exactly three ASNs, we have $|\langle V, \mathcal{A}, T, \mathcal{B}, \mathcal{S}\rangle| = 3$, where $V = T$, $\mathcal{A} = \emptyset$, $|\mathcal{B}| = 1$, and $|\mathcal{S}| = 1$. In Figure 2, $V \neq T$ and are shown explicitly, $\mathcal{A}$ is implicit, and $\mathcal{B} = \langle N_3, N_4, N_5\rangle$. The blue line in Figure 2 captures three routes from origins N_6, N_7, and N_8; the suffix $\mathcal{S}$ of each route contains only the origin AS.

We denote by $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ the set of routes from one specific vantage point that traverse the sequence of ASes given by $\langle V, \mathcal{A}, T, \mathcal{B}\rangle$. Each route $r \in \mathcal{R}$ has a different nonempty suffix $\mathcal{S}_r$. Table 1 summarizes the notation, and Algorithm 1 shows the pseudocode.

3.2.1 Minimum number of origins. For any combination of V, $\mathcal{A}$, T, and $\mathcal{B}$ from each vantage point, we consider the set of routes $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ for inferring location communities if $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ contains at least K_{origins} distinct routes. In other words, we require announcements from at least K_{origins} distinct origin ASes to avoid incorrect inferences when origin ASes tag all their announcements with AS T traffic engineering communities (Lines 3–6 in Algorithm 1.)

3.2.2 Community prevalence. One could require a BGP community from the target AS T to appear on *all* routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ in order to infer it as a location community. However, Internet routing information is often incomplete or inconsistent, *e.g.,* due to delayed route propagation [34] or ASes that remove BGP communities from announcements.[2] Rather than requiring a community to appear on all routes, we relax this requirement to allow for incompleteness and inconsistency in BGP dumps or route propagation, and infer any community from AS T or its siblings that appears on at least a fraction K_{prev} of routes in $\mathcal{R}$ as a location community (Lines 7–13 in Algorithm 1).

We have observed that ASes often tag routes with location communities of a sibling AS. Sibling ASes are operated by the same organization and often share routes and operational practices [13, 14]. Sibling ASes may share BGP communities to avoid defining and maintaining multiple sets of communities for different ASes belonging to the same organization. Sharing of BGP communities may also happen during mergers, when updating AS numbers requires reconfiguration of BGP sessions and coordination with peering networks. As an example, in 2018, after Level3's and Global Crossing's merger [8, 31], we observed routes with AS paths traversing Global Crossing's AS3549

[2]BGP communities are a transitive attribute and ASes are not supposed to arbitrarily remove them from routes [5]. However, filtering of BGP communities is available as a router configuration option from most vendors. Recent work reports that 25% of ASes filter communities from routes [37, 38].

Algorithm 1 Inference of Location Communities

```
1: for each vantage point v do
2:   L_v ← ∅ {Set of location communities inferred from v's routes}
3:   for each R(V, A, T, B) in routes from v do
4:     if |R(V, A, T, B)| < K_origins then
5:       continue
6:     end if
7:     C ← all communities from AS T or of a sibling of T appearing in R(V, A, T, B)
8:     for each community c ∈ C do
9:       N_c ← number of routes in R(V, A, T, B) with c
10:      if N_c ÷ |R(V, A, T, B)| ≥ K_prev then
11:        L_v ← L_v ∪ {c}
12:      end if
13:    end for
14:  end for
15:  for each community c ∈ L_v do
16:    R_c ← set of routes with c
17:    R_T ← set of routes whose AS paths traverse c's AS or any of its siblings
18:    F_c ← R_c \ R_T
19:    if size of the minimum hitting set of F_c ≥ K_filter then
20:      L_v ← L_v \ {c}
21:    end if
22:  end for
23: end for
24: return ⋃ L_v for all vantage points v
```

tagged with Level3's location communities. Thus, when processing $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$, we try to infer communities from T or any of T's siblings as location communities (Line 7, Algorithm 1).

3.2.3 Removing communities unrelated to location. We develop a heuristic to filter out BGP communities that are unlikely to be location communities. We expect a location community to be tagged when an AS receives a route. Thus, a location community from AS T should only appear on routes whose AS path includes AS T or one of its siblings.

Unfortunately, databases identifying sibling ASes are challenging to build and may be incomplete, leading direct application of the heuristic to incorrectly discard inferred location communities. For example, we observed several routes traversing AS286 and AS5580 tagged with location communities from GTT's AS3257. Manual querying of ARIN's IRR indicates that these three ASes are siblings, but they are not identified as such in CAIDA's sibling database (Section 4).

Another issue is that there are ASes that seem to tag routes with location communities of other ASes, with no apparent sibling relationship. For example, we observed announcements traversing AS20473 (Constant) tagged with location communities from AS1299 (Telia).[3]

We relax the heuristic to allow for missing sibling ASes and ASes that reuse or incorrectly tag announcements with another AS's location communities. We try to identify cases where a small set of ASes can be blamed for the tagging of a target AS T's communities on routes that do not traverse T or any of T's known siblings. In these cases, we do *not* filter out inferred location communities.

[3] Although we could not establish a sibling relationship between AS20473 and AS1299, we plan to investigate this further as BGP community cross-tagging might be a possible vector for identifying sibling ASes.

More precisely, let $\mathcal{R}_c$ be the set of routes tagged with community c from AS T ($\mathcal{R}_c$ is a superset of, and usually much larger than, the set $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ used to infer c as a location community), and let $\mathcal{R}_T$ be the set of routes whose AS paths traverse AS T or any of T's known siblings. We ignore routes that traverse T or any of T's siblings, and consider the route announcements $\mathcal{F}_c = \mathcal{R}_c \setminus \mathcal{R}_T$ when deciding whether to discard an inferred location community. We compute the *minimum hitting set* of $\mathcal{F}_c$ and discard c as a location community if the set contains more than K_{filter} ASes (Lines 15–22 in Algorithm 1). In other words, we keep location community inferences only when few ASes are to blame for AS T's communities showing up on routes that do not contain T or any of T's siblings. The minimum hitting set is the smallest set of ASes $\mathcal{W}$ such that the intersection of $\mathcal{W}$ and each route $r \in \mathcal{F}_c$ is nonempty. The minimum hitting set problem is equivalent to the NP-complete minimum set cover problem [15, 33], and we solve it using a greedy heuristic, which provides a tight approximation of the optimal solution [52].

3.2.4 Joining inferences across collectors. We infer location communities from route announcements observed by each vantage point in isolation (loop in Line 1, Algorithm 1), in line with the ideas of using each BGP collector as a vantage point and $\mathcal{B}$ to constrain where chosen routes are received. After we infer communities from each vantage point, we take the union across vantage points from all collectors as the database of inferred location communities (Line 24 in Algorithm 1). Although we show that few vantage points are sufficient to infer most communities (§5), some communities are only visible from specific vantage points, so taking the union across collectors and vantage points maximizes coverage.

3.3 Implementation

Our implementation consists of over 2,100 lines of Python, with extensive use of the Pandas library for data processing. We use Snakemake [43] to automate our database construction. Our system can be configured to automatically process multiple RIBs from different BGP collectors, generate various intermediate files that are reused in subsequent steps, and distribute the processing into multiple servers to speed up the computation. Our code, the database of inferred communities, and our manually built ground-truth dataset are available online [32].

4 DATASETS

We use BGP feeds from RouteViews [42], RIPE RIS [47] and Isolario [25].[4] Unless specified otherwise, we use the first available route table dump (RIB) from each BGP route collector on December 2017, 2018, 2019, and 2020. We use BGP RIBs to process stable routes, but BGP updates could also be used, which would possibly increase the number of observed communities. Table 2 shows a summary for the route table dumps from December 2017 and 2020. We use CAIDA's AS-to-Org database for identifying sibling ASes [3]. We built and evaluated an alternate sibling database by grouping ASes whose abuse contact e-mails have the same domain. We omit these results as they are quantitatively similar to those obtained with CAIDA's AS-to-Org database. When processing routes, we remove repeated occurrences of an ASN in the AS path as our goal is to look at the sequence of ASes traversed by the route regardless of AS path prepending. We also discard all routes containing AS-sets, as they usually result from aggregation of routes traversing different ASes.

Table 3 shows a summary of our manually-built ground-truth dataset of BGP community semantics for ASes that have public information available. We obtain ground-truth information from IRR databases and documentation from network websites, and manually classify each community on June 2021. The ground-truth dataset contains a large number of communities because some ASes specify certain types of communities using ranges, and we consider all possible values defined

[4] We do not use PCH feeds [28] as they do not include BGP communities.

Table 2. Summary of RIB dumps of December 2017 and 2020 for RouteViews, RIPE RIS, and Isolario.

Project	Collectors		VPs		Total ASes (thousands)		Prefixes (millions)		Communities (thousands)		Routes (millions)	
Year	2017	2020	2017	2020	2017	2020	2017	2020	2017	2020	2017	2020
RV	17	20	192	232	61	72	0.86	1.07	44	64	96	184
RIPE	20	20	330	510	61	72	0.80	1.03	46	71	115	311
Isolario	4	5	83	145	60	72	0.79	1.12	34	67	66	209
Total (unique)	41	45	529	738	61	73	0.90	1.22	56	79	277	704

Table 3. Number of communities for ASes in our ground-truth dataset by type and geolocation communities in CAIDA's database [4].

	Community Type				CAIDA
Network (AS)	Geo	Dev/link	Relation	Action	[4]
Tier 1 [60]					
Verizon (701)	0	0	0	11	0
NTT (2914)	93	0	2	44	39
GTT (3257)	10,000*	11,000*	1,783*	13,023*	68
Deutsche Telekom (3320)	24	0	3	0	17
Level 3 (3356)	178	0	2	5	82
PCCW Global (3491)	44	0	0	21	24
Lumen (3549)	239	239	239	87	28
Orange (5511)	46	0	0	55	11
Zayo (6461)	804*	0	6	152	0
Telecom Italia (6762)	51	0	1	133	42
Tier 2 [61]					
Cogent (174)	4	0	0	47	31
TDC (3292)	0	0	3	119	12
Easynet (4589)	800*	0	0	3	103
British Telecom (5400)	0	0	0	40	0
Comcast (7922)	0	0	0	7	0
Total					
	12,283	11,239	2,039	13,747	457

* Ranges covering automatically-generated community values, *e.g.*, from geographical coordinates.

in the range (although our evaluation indicates actual utilization is sparse). For example, GTT (AS3257) defines a rule saying that communities in the 3257:30000–3257:39999 interval identify private interconnections [53]. In this case, we consider all 10,000 communities in the interval as location communities in our ground-truth database.

We break informational communities into those that identify a geographical location, a device, a link on a router, or a peering relationship; and we also identify action communities. We also show the number of communities from the ASes in our database in CAIDA's geographic location BGP communities database from April 2019 [4]. Our ground truth dataset includes 1.7 times more geolocation communities than CAIDA's database for the ASes in our dataset (not including autogenerated communities), but 13% of the geolocation communities in CAIDA's database are not in our ground-truth dataset.

Manual analysis indicates that these differences are due to new geolocation communities being created since CAIDA's database was built, and a few changes to previously-assigned ones.

5 EVALUATION

In this section we evaluate our algorithm. We report precision and recall, and show how they can be prioritized by tuning the configuration of our algorithm (§5.1). We discuss community visibility in BGP dumps and how additional vantage points could improve recall (§5.2). We quantify the

Table 4. Precision, recall, inferable recall, F1-score, and the number of inferred, correctly inferred, and inferred but undocumented location communities on December 2020. We show results for our algorithm's default configuration as well as configurations that prioritize high precision and high recall.

			Inferable		Inferred Communities		
Configuration	Precision	Recall	Recall	F1 Score	Total	Correct	Undocumented
Prioritize precision	0.93	0.72	0.89	0.81	946	878	513
Default configuration	0.91	0.80	0.87	0.85	1081	983	598
Prioritize recall	0.87	0.81	0.89	0.84	1150	995	634

impact of each parameter on our algorithm's accuracy and show that inferences are not sensitive to parameter values (§5.3). We compare our database of location communities with CAIDA's manually-built dataset and show we achieve competitive precision and significantly higher recall (§5.4). Finally, we present a characterization of the adoption and stability of location communities (§5.5).

5.1 Inference Accuracy

We quantify inference accuracy with precision and different views of recall [27]. *Precision* is the ratio between the number of correctly inferred location communities (true positives) and the number of inferred communities (positives). As our ground-truth database contains many communities that are not yet used (*i.e.,* communities described as ranges on the providers' websites but not yet allocated), it would be unreasonable to use them to compute the recall. Furthermore, many communities that are defined in the ground-truth dataset never show up in BGP dumps, possibly because they are not in use or because vantage points lack visibility. We compute *recall* considering only the communities that appear in the BGP dumps. More precisely, we define recall as the ratio between true positives and the number of location communities in our ground-truth database that also appear in the BGP table dumps. We also report the *inferable recall*, defined as the ratio between true positives and the number of communities that our algorithm considers for inference, *i.e.,* communities that appear on routes from at least K_{origins} origin ASes.

Table 4 shows the overall accuracy of our inferences for its default configuration, with $K_{\text{origins}} = 2$, $K_{\text{prev}} = 0.2$, and $K_{\text{filter}} = 1$ on December 2020. We evaluate the impact of each parameter and discuss the default choices in Section 5.3. Table 4 also reports the *total* number of inferred communities across ASes in our ground-truth dataset, the number of *correctly* inferred location communities, and the number of inferred location communities that are *undocumented* in the ground truth. Communities may be undocumented in the ground-truth because they are meant for private use of the owning AS, or may be incorrectly tagged on routes. Because we cannot know whether the inferences for undocumented communities are correct or incorrect, we ignore them when computing precision and recall.

Our results show that inference precision is high. We find that 34.3% of location communities in the ground-truth that are not auto-generated never appear in the BGP dumps, which makes inference impossible. However, we do find reasonably high recall for observed communities. Results for configurations prioritizing high precision ($K_{\text{origins}} = 6$, $K_{\text{prev}} = 0.5$, and $K_{\text{filter}} = 1$) and high recall ($K_{\text{origins}} = 2$, $K_{\text{prev}} = 0.1$, and $K_{\text{filter}} = 2$) indicate that our algorithm can be configured to trade off precision against recall depending on the operator's, researcher's, or application's needs.

Table 5 shows the breakdown of the number of communities per category. The *seen* columns show the number of communities in the BGP dump and in our ground-truth dataset, and the *inferred* columns show the number of communities we infer as location communities. Despite an imbalanced dataset and the high number of false positives for action communities, our algorithm would still yield a *positive predictive value* [57] of 79% even if location and action communities

Table 5. Number of communities from ASes in our ground-truth dataset *seen* in BGP and *inferred* as location communities by our algorithm. We split communities by type, as given in the ground truth (*location*, *relationship*, and *action*), and also show results for *undocumented* communities that do not show up in the ground truth.

	Community Category							
	Location		Relationship		Action		Undocumented	
Configuration	seen	inferred	seen	inferred	seen	inferred	seen	inferred
Prioritize precision	987	878	14	13	181	55	675	513
Default configuration	1123	983	15	13	235	85	911	598
Prioritize recall	1123	995	15	15	235	140	911	634

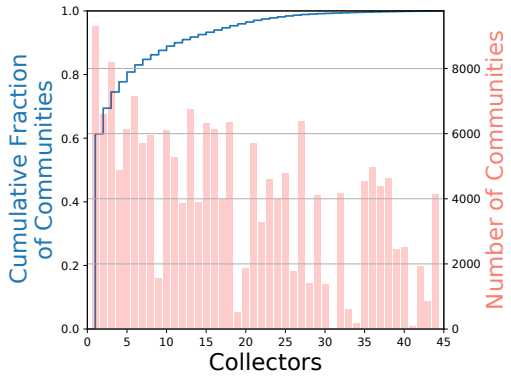


Fig. 3. Number of inferred communities and recall as a function of the number of collectors.

were balanced.[5] We can increase the precision for action communities by tuning the algorithm's parameters (*e.g.*, prioritize precision). We discuss these limitations and future work in Section 6.

5.2 Community Visibility and Recall

Figure 3 shows the cumulative distribution of the number of inferred location communities (left y-axis) and the number of inferred communities (right y-axis) across collectors (x-axis). We rank collectors on the x-axis by picking the collector that supports the most inferences, and then iteratively selecting collectors by the number of new community inferences they support. Note that we can infer a large number of communities in one collector, but those communities might have already been inferred in a previous collector. That explains why we see some shorter bars on the left of higher ones. For example, we inferred 17 communities from routes exported by vantage points connected to the collector at rank 31, and 13 of those communities were new, while we inferred 4,525 communities from routes exported by vantage points connected to the collector at rank 35, and only 10 communities were new.

The number of inferred communities varies significantly across collectors, which can be explained by the different number of vantage points. We observe correlation (Pearson correlation coefficient of 0.7) between the number of vantage points of a collector and the number of inferred communities (not shown).

We also find that there is significant overlap among communities inferred from different collectors. This explains why the fraction of inferred communities spikes to 61% with a single collector, and

[5]This ignores relationship communities, which we expect to be few and not balanced, as an AS generally defines *one* community for each type of relationship (provider, peer, or customer).

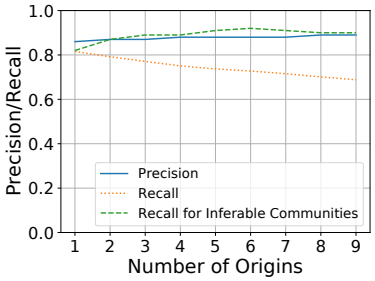


Fig. 4. Precision and recall as a function of K_{origins}. High precision for $K_{\text{origins}} = 1$ indicates that origins rarely tag *all* their announcements with traffic engineering communities of other ASes.

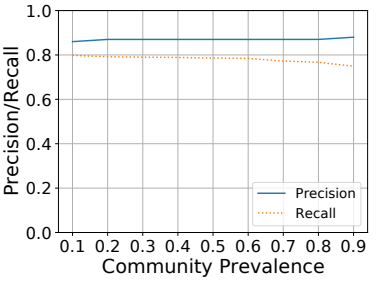


Fig. 5. Precision and recall as a function of K_{prev}. Location communities appear on most routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$, so increasing K_{prev} up to 0.9 has small impact on precision and recall.

then grows slowly. However, even though the growth is slow as a function of the number of collectors, the tail of the distribution is long, indicating that some communities can only be inferred by specific vantage points.

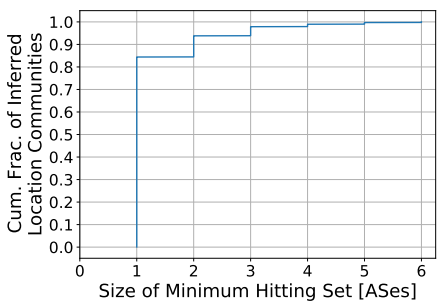
These results indicate that additional collectors and vantage points would allow inferences to achieve higher coverage and recall, but that the existing set of collectors is sufficient to enter the region where additional collectors will provide diminishing returns on community visibility.

5.3 Algorithm Parametrization

In this section we quantify the impact of configuration parameters in our algorithm. Our results show that our algorithm is not sensitive to parametrization and that most parameter values yield accurate predictions.

5.3.1 Number of origins. Figure 4 quantifies the impact of K_{origins} on precision and recall. We observe that precision increases slightly with K_{origins} as we require routes from more diverse origins. One factor contributing to improving precision is that larger K_{origins} makes the algorithm less susceptible to incorrect inferences when origins tag all their announcements with another ASes's traffic engineering communities. However, we observe that recall decreases as K_{origins} increases. This happens because the number of routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ traversed by K_{origins} distinct origins decreases, and thus the number of routes useful for inferring communities also decreases. The limited improvement in precision implies that origins rarely tag *all* their announcements with traffic engineering communities of other ASes. We argue that any choice of K_{origins} is reasonable as it trades off precision and recall. Values of K_{origins} larger than one have the advantage of avoiding incorrect inferences in situations where an AS tags all its routes with traffic engineering communities. We choose $K_{\text{origins}} = 2$ as the default value in our algorithm as an intermediate value that prevents a single origin causing incorrect inferences without significantly degrading recall.

Figure 4 also shows the recall of *inferable communities*, *i.e.,* communities from ASes in AS path segments shared by at least K_{origins} origins. This is relevant because we cannot make inferences for communities that do not appear in paths from enough different origins. We find that recall for inferable communities increases with K_{origins}, indicating that our algorithm performs better on communities that appear on paths shared by many origins, which may be a result of a lack of path diversity from these origins towards the target AS T, funneling traffic through fewer locations.



(a) Distribution of minimum hitting set sizes

(b) Precision and recall as a function of K_{filter}

Fig. 6. Most inferred location communities appear on routes traversing the community's controlling AS or one of the controller AS's siblings (not shown). For 85% of the inferred location communities that appear on routes that do *not* traverse the controlling AS or one of its siblings, we find that a single AS can be blamed for tagging the community (Figure 6a, $x = 1$). Filtering inferences when a community appears on a diverse set of routes that do not traverse the controlling AS or one of its siblings improves the precision of our inferences without significantly reducing recall (Figure 6b).

5.3.2 Community prevalence. Figure 5 shows the impact of K_{prev}, the fraction of routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ that a community needs to appear in to be inferred as a location community. Similar to Figure 4, we find that precision and recall are high and do not vary significantly as a function of K_{prev}. This happens because *(i)* location communities have high prevalence, so increasing K_{prev} has small impact on the number of true positives, and *(ii)* other communities have low prevalence and get promptly filtered as we increase K_{prev} from zero. We set $K_{\text{prev}} = 0.2$ as the default value in our inferences, *i.e.,* we require that a community appears in at least 20% of the route announcements in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ tuple to infer it as a location community.

5.3.3 Filtering inferences. We filter the inference of an AS T's community from our database of location communities if it appears on paths that do not traverse T or any of T's siblings and the appearances cannot be blamed on K_{filter} or fewer ASes.

Figure 6a shows the distribution of the number of ASes in minimum hitting sets for inferred communities. We observe that the majority of hitting sets (85%) have only one AS, which implies that a single AS can be blamed for occurrences of those communities on paths that do not traverse the community's AS (or any sibling). A possible explanation for this finding is that these single ASes may be undocumented siblings of the community's AS or may incorrectly tag routes with the community. Figure 6b shows the impact of K_{filter} on precision and recall. We plot the x axis for decreasing values of K_{filter} as the filter becomes more restrictive (*i.e.,* we infer fewer location communities) as K_{filter} decreases. The results show that values of K_{filter} below 3 have a slight impact on precision, without impacting recall. This indicates that the proposed filter accurately identifies and prunes incorrect inferences. We set the default value of $K_{\text{filter}} = 1$ in our algorithm.

We also quantify how often ASes use communities from one of their siblings. We say an AS A uses a community from its sibling AS T when a community owned by T appears in an AS-path that includes A and does not include T. We find 95 ASes using communities defined by their siblings in BGP dumps (across all ASes and all communities regardless of semantics), and our algorithm infers location communities for 44 of these ASes. This result indicates that siblings do share BGP communities, and accounting for this sharing is useful when filtering location communities.

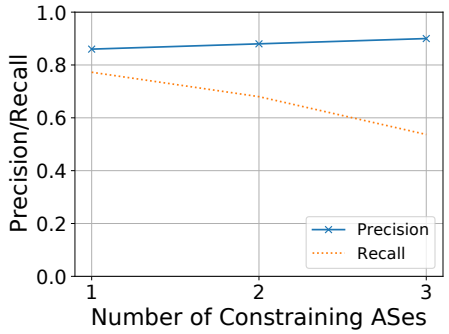


Fig. 7. Impact of the number of constraining ASes, *i.e.,* $|\mathcal{B}|$, on recall and precision. More constraining ASes limit where chosen paths are received by a target AS T, improving precision, but fewer AS paths are long enough to support many constraining ASes, reducing recall.

5.3.4 Number of constraining ASes. Figure 7 shows the impact of the number of constraining ASes after T in the AS path when making inferences, *i.e.,* the size of $\mathcal{B}$ in $\langle V, \mathcal{A}, T, \mathcal{B} \rangle$ tuples. As discussed in Section 3.1, more constraining ASes limit the set of locations where chosen routes arrive at the target AS T, leading to higher precision. However, AS paths in the Internet are usually short [6], and there are fewer long AS paths to support inferences with long sequences of constraining ASes, which ultimately limits recall. Although we consider all sequences with at least one constraining AS, our algorithm can be configured to require more constraining ASes, which will lead to higher precision at the cost of recall.

5.4 Comparison with CAIDA's Database

We now turn to properties of our inference algorithm and compare the constructed database with CAIDA's public database. Table 6 shows statistics for geolocation communities in both databases (first rows) and for location communities in our database (last row). We compute recall of geolocation communities considering only the subset of geolocation communities in the ground-truth database. We do not compute precision and the number of geolocation community for our inference algorithm as it does not differentiate between geolocation and location communities.

We find that CAIDA's database has high precision, but not 100%. Investigation of incorrect inferences indicate they are concentrated on Tier-2 ASes and explained by out-of-date information, *e.g.,* resulting from the reassignment of community values. Also, CAIDA's community database has limited recall, which is somewhat expected for a manually-built database. Our inference algorithm achieves significantly higher recall than CAIDA's database even for geolocation communities.

The last row shows results for all location communities inferred by our algorithm. We find that recall increases slightly compared to when we consider only geolocation communities. We also find that the precision is competitive with that of manually-constructed but not up-to-date databases.

5.5 Adoption and Stability of Location Communities

Figure 8 shows the number of distinct BGP communities observed in the BGP route dumps, the number of communities inferred as location communities, the number of ASes covered in the BGP route dumps, and the number of ASes controlling the observed communities. We find that BGP communities are becoming more popular, with a 51% increase in the number of distinct communities

Table 6. Comparison between CAIDA's manually-constructed database and our automatic inferences.

Community Type	Database	Recall	Precision	Communities Total	Communities Correct
Geolocation	CAIDA	0.21	0.86	303	261
	Inferences	0.77	—	—	—
Location	Inferences	0.80	0.91	1081	983

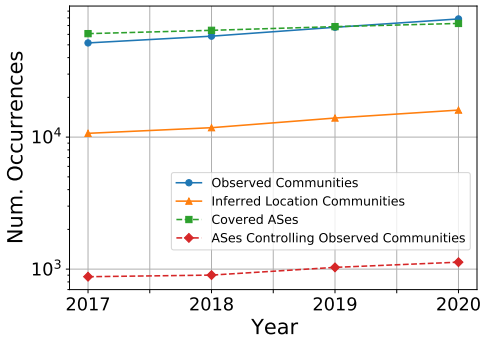


Fig. 8. BGP community use in the Internet, quantified as the number of distinct BGP communities observed, number of inferred location communities, and the number of ASes controlling BGP communities.

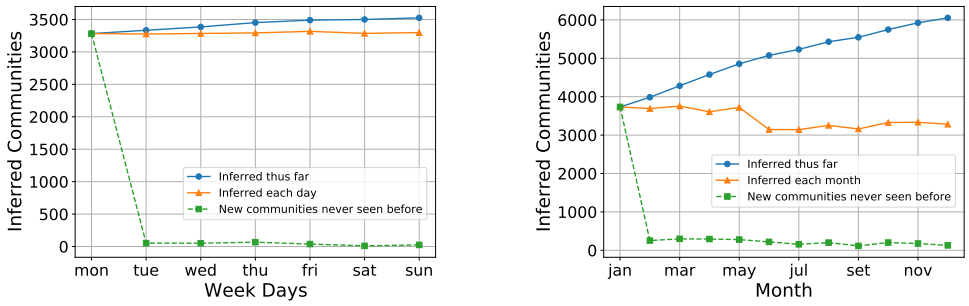
observed in the wild between 2017 and 2020 (50% increase for location communities). Not only are there more communities, but they also belong to a larger number of ASes.

Figure 9 evaluates how stable are location community inferences over time. Figure 9(a) shows the total number of communities inferred each day over the course of the first week of December 2020. We report the number of new communities never seen before (green line), the number of inferences on each day (orange line), and the cumulative number of communities inferred (blue line). We find that the set of inferred communities does not change significantly over the course of one week. Figure 9(b) is similar, but shows communities inferred on the first day of each month in 2020. We find that there is some stability, but distinct communities keep accumulating over time. This result can be explained by changes in topology accompanied by the creation of new location communities, *e.g.,* when networks establish PoPs in new locations, or routing dynamics, *e.g.,* new peering relationships may lead to route changes that allow the inference of new location communities. The change over time motivates an automated algorithm like the one we propose for keeping the community database up-to-date. The drop in the number of inferred communities around June 2020 can be mostly attributed to the disappearance of AS286's communities from BGP dumps, likely a result of AS286's acquisition by GTT (AS3257) in December 2019.

6 LIMITATIONS AND FUTURE WORK

We require announcements from K_{origins} distinct origin ASes to sidestep the case that an origin AS tags all its announcements with traffic engineering communities of T. Unfortunately, if any AS in $\mathcal{B}$ tags *all* announcements from *all* origins with one of T's traffic engineering communities, then our algorithm would incorrectly infer a traffic engineering community as a location community.

Our algorithm will also falsely infer a location community when AS T tags all received routes from a neighbor with a relationship community (*e.g.,* peer, customer, or provider). However, the decrease in precision is not significant because an AS generally defines a small number of relationship



(a) First week of December 2020

(b) First day of each month in 2020

Fig. 9. Stability of location community inferences over time. Our results show that location communities are stable over short timescales, but that new location communities appear over time. This motivates an automated inference algorithm to keep community databases up-to-date.

communities, as we show in Section 5. We can avoid this case by requiring that a community appears in routes from neighbors with different relationships at the expense of a lower recall. We will investigate this trade-off in future work.

Our algorithm is unable to make inferences when ASes between the BGP collector and the target AS T remove communities from BGP announcements [38]. As the number of collectors on the Internet is large, which provides visibility from multiple vantage points, we believe (based on the results in Section 5) that we can still achieve high recall even if some ASes remove the communities.

We built our ground-truth database from Tier-1 and Tier-2 ASes because we could not find documented communities for ASes lower in the Internet hierarchy. However, we expect our techniques to work well for any target AS T as long as routes received by T have at least $K_{origins}$ and a set $\mathcal{B}$ with at least one AS, as we discuss in Section 5.3.4.

We believe other heuristics may further improve our ability to infer the semantics of BGP communities. We plan to investigate whether we can identify traffic engineering communities (e.g., "do not announce in Europe") correlating changes in AS-paths to specific BGP communities. We expect a better understanding of the semantics of BGP communities will support new solutions; for example, we plan to investigate whether announcements where AS x uses another AS y's location communities can be used to infer whether ASes x and y are siblings.

7 RELATED WORK

Characterization of community usage. Streibelt *et al.* [56] present an extensive study of BGP community usage on the Internet. The study shows the growing use of communities in the last few years and how communities propagate much further than they should, sometimes reaching ASes several hops away from the intended target. Unintended forwarding of communities to upstream neighbors allows adversaries to trigger remote blackholing to disconnect destinations or to influence route propagation to steer traffic through malicious actors without resorting to a prefix hijack. The authors argue that standardization and better documentation of BGP communities could prevent such abuses, which our paper is one step towards. Krenc *et al.* [38] propose a passive algorithm to infer how ASes handle communities. BGP communities are a transitive attribute of BGP updates, which means they should propagate from one AS to the next; however, routers may be configured to filter them. Krenc *et al.*'s algorithm infers whether an AS forwards or discards communities in BGP announcements. Their algorithm, like ours, also uses only passive measurements from BGP collectors to infer the different types of ASes. We note that, although filtering of BGP communities reduces recall of our algorithm, it does not impact precision of inferences.

Standardization efforts. Quoitin and Bonaventure [44] discuss the two most common utilization of BGP communities in the Internet: communities that tag routes received from a specific peer (*e.g.,* to indicate the type of relationship) or at a specific location, and communities that indicate how external peers should redistribute a route to perform, for example, interdomain traffic engineering. Donnet and Bonaventure [11] extend the classification in [44] and propose a taxonomy of BGP communities that allows network operators to document their communities better. The first level of the taxonomy divides the communities into inbound, outbound, and blackhole. Subsequent levels specialize their applications into several categories: tagging, route redistribution, type of peer, IXP, geographic, and prepend. Birge-Lee *et al.* [2] augment the taxonomy in [11] and present an extensive discussion about where communities should not be accepted (*e.g.,* ASes should not accept communities from peers or providers) or propagated (*e.g.,* community propagation should be limited to two hops). These measures, however, can significantly limit legitimate uses of communities. Unfortunately, network operators have not embraced the proposed taxonomies, challenging the development of automated tools for cataloging existing communities.

Inference of community semantics. Recent efforts use natural language processing (NLP) to automatically identify the semantics of BGP communities from Internet Routing Registries and support webpages of network providers [18, 21]. These data sources are generally incomplete and outdated, significantly limiting the number of communities that approaches based on NLP can achieve. On the other hand, our approach automatically generates an up-to-date database contains BGP communities currently in use by the network operators, increasing coverage and precision.

Legitimate uses of BGP communities. Determining the relationship between two ASes is a hard problem, but it has many applications [41]. In particular, network operators can detect if route announcements do not violate practical norms, such as advertising routes from a peer to a provider, that may lead to route leaks and disrupt the traffic of large portions of the Internet. Giotsas *et al.* [18] shows that a reliable dictionary of BGP communities can significantly improve the detection of infrastructure outages, Feldman *et al.* [12] use communities to locate routing instabilities, and Giotsas *et al.* [19] look for changes in communities to identify intradomain path changes.

Malicious uses of BGP communities. Some works have shown that BGP communities can be a vector for malicious attacks [2, 56]. Interception attacks based on prefix hijacks generally disrupt significant portions of the Internet [49], which induces quick detection and remediation by network operators. SICO [2] builds community-based interception attacks that target small portions of the Internet and are harder to detect. Streibelt *et al.* [56] present several scenarios where a malicious actor can abuse BGP communities to launch several types of attack, as we mentioned above. These attacks generally rely on action communities, such as the blackhole and no-export communities, and improperly configured routers that forward non-transitive communities. While location communities, the focus of our work, can improve route visibility, their use as an attack vector is limited as they do not directly trigger any action on a remote network.

Inference of AS relationships. Autonomous Systems connect to each other and exchange routes based on business relationships. AS relationships can be broadly classified into four categories: customer-to-provider (c2p), provider-to-customer (p2c), settlement-free peering (p2p), and sibling-to-sibling (s2s). Unfortunately, these business relationships are rarely disclosed, which reduces the amount of metadata available to annotate the Internet's AS graph, and consequently complicates the deployment of many applications such as congestion detection between ASes with specific peering agreements [9] (*e.g.,* congestion on an ISP link to a client), malicious AS identification, and deployment of BGP security mechanisms [17, 36, 51]. For the past 20 years, researchers have proposed different techniques to infer AS relationships [13, 20, 24, 30]. Most techniques assume

that BGP paths follow the *valley-free* property, which states that a path is a sequence of zero or more c2p links, followed by zero or one p2p link, and zero or more p2c links [13]. Jin *et al.* and Giotsas *et al.* [20, 30] argue that AS relationships are more complex and propose algorithms to infer non-conventional peering practices, such as hybrid relationships, in which ASes have different relationships depending on the peering location, and non-valley-free routing resulting from sibling relationships. A few efforts [44, 59, 62] propose or discuss the use of BGP communities to infer AS relationships and show that they enable great accuracy. Our algorithm uses CAIDA's AS-to-Org database [3] to detect sibling ASes, but does not rely on AS relationship inferences. A more complete database of sibling ASes could improve the precision of inferences.

8 ETHICAL CONCERNS

To build our community database, we use publicly available datasets voluntarily exported to BGP collectors by autonomous systems on the Internet. Our techniques do not send active probes.

Location communities are informational communities that do not trigger any action on peering or remote ASes. The known reported attacks using BGP communities rely exclusively on action communities [2, 56]. Furthermore, our database lists only the semantics of the communities and not the specific geographic locations they represent, so an attacker would have to glean complementary information from diverse data sources to plan a targeted attack.

Our community database will be valuable for network operators and researchers to reason about traffic dynamics on the Internet, improve network performance, and check policy compliance. We believe that the positives of a public database of location communities far outweigh the possibility of misuse for malicious activities.

9 CONCLUSION

In recent years, the use of BGP communities has increased significantly. As routing policies have become more complex and performance requirements have become more stringent on the Internet, network operators have to deploy ever more elaborate traffic engineering solutions. Traffic engineering solutions can utilize information and action BGP communities to achieve operational goals, and our results indeed indicate an uptick in the adoption of BGP communities. Unfortunately, there is no standard for specifying semantics nor a centralized repository that catalogs BGP communities, which complicates their use by network operators and researchers.

Our work is the first we are aware of to use routing announcements publicly available from BGP collectors to infer the semantics of BGP communities. We leverage the existing routing BGP collectors as a positioning system to correlate route announcements with the locations that a route traverses. Our algorithm automatically infers location communities and achieves high precision (93%) and recall (81%) for communities from a set of Tier-1 and Tier-2 ASes. Compared with the manually built database from CAIDA [4], our inference algorithm generates a database with similar precision and much higher recall. We make our database with 15,505 inferred location communities as well as our code publicly available [32].

ACKNOWLEDGMENTS

We would like to thank our shepherd Oliver Hohlfeld and the anonymous ACM SIGMETRICS reviewers for their valuable feedback. This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, CAPES – Finance Code 001, and FAPESP procs. 14/50937-1 and 15/24485-9. This work was also supported in part by the Brazilian National Research and Educational Network (RNP) convs. 2955 and 2956, FAPESP proc. 2020/05192-9, CNPq project 432339/2018-3, FAPEMIG proc. APQ-02856-18, and NSF Award 1740883 (under a joint US-Brazil Collaboration on Cybersecurity Research).

REFERENCES

[1] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. 2015. Investigating Interdomain Routing Policies in the Wild. In *Proceedings of the 2015 Internet Measurement Conference*. ACM, Tokyo, Japan, 71–77. https://doi.org/10.1145/2815675.2815712

[2] Henry Birge-Lee, Liang Wang, Jennifer Rexford, and Prateek Mittal. 2019. SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. ACM Press, London, United Kingdom, 431–448. https://doi.org/10.1145/3319535.3363197

[3] CAIDA. 2021. CAIDA's AS-Organization Dataset. http://data.caida.org/datasets/as-organizations/. [Online; accessed 11-Oct-2021].

[4] CAIDA. 2021. CAIDA's Geolocation Dataset. https://www.caida.org/catalog/datasets/bgp-communities/. [Online; accessed 11-Oct-2021].

[5] R Chandra, P Traina, and T Li. 1996. *BGP Communities Attribute.* Technical Report. RFC 1997, August.

[6] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, and Ramesh Govindan. 2015. Are We One Hop Away from a Better Internet?. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference (IMC '15)*. ACM Press, Tokyo, Japan, 523–529. https://doi.org/10.1145/2815675.2815719

[7] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. 2019. BGP Hijacking Classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, Paris, France, 25–32. https://doi.org/10.23919/TMA.2019.8784511

[8] Federal Communications Commission. 2021. FCC: Global Crossing and Level 3 Proposed Merger. https://www.fcc.gov/proceedings-actions/mergers-transactions/global-crossing-ltd-and-citizens-communications-company [Online; accessed 11-Oct-2021].

[9] Amogh Dhamdhere, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky KP Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and Kc Claffy. 2018. Inferring Persistent Interdomain Congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. ACM, Budapest, Hungary, 1–15. https://doi.org/10.1145/3230543.3230549

[10] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, KC Claffy, and George Riley. 2007. AS Relationships: Inference and Validation. *ACM SIGCOMM Computer Communication Review* 37, 1 (2007), 29–40. https://doi.org/10.1145/1198255.1198259

[11] Benoit Donnet and Olivier Bonaventure. 2008. On BGP Communities. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 55–59.

[12] Anja Feldmann, Olaf Maennel, Z. Morley Mao, Arthur Berger, and Bruce Maggs. 2004. Locating Internet Routing Instabilities. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Portland, Oregon, USA) *(SIGCOMM '04)*. ACM, New York, NY, USA, 205–218. https://doi.org/10.1145/1015467.1015491

[13] Lixin Gao. 2001. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking* 9, 6 (2001), 733–745. https://doi.org/10.1109/90.974527

[14] Lixin Gao and Jennifer Rexford. 2001. Stable Internet Routing Without Global Coordination. *IEEE/ACM Transactions on Networking* 9, 6 (2001), 12. https://doi.org/10.1109/90.974523

[15] Michael R. Garey and David S. Johson. 1979. *Computers and Intractability – A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, NY.

[16] GBLX. 2021. GBLX Customer BGP Communities. https://onestep.net/communities/as3549/

[17] Phillipa Gill, Michael Schapira, and Sharon Goldberg. 2011. Let the Market Drive Deployment: A Strategy for Transitioning to BGP Security. In *Proceedings of the ACM SIGCOMM 2011 Conference* (Toronto, Ontario, Canada) *(SIGCOMM '11)*. Association for Computing Machinery, New York, NY, USA, 14–25. https://doi.org/10.1145/2018436.2018439

[18] Vasileios Giotsas, Christoph Dietzel, Georgios Smaragdakis, Anja Feldmann, Arthur Berger, and Emile Aben. 2017. Detecting Peering Infrastructure Outages in the Wild. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, New York, NY, USA, 446–459. https://doi.org/10.1145/3098822.3098855 event-place: Los Angeles, CA, USA.

[19] Vasileios Giotsas, Thomas Koch, Elverton Fazzion, Ítalo Cunha, Matt Calder, Harsha V Madhyastha, and Ethan Katz-Bassett. 2020. Reduce, Reuse, Recycle: Repurposing Existing Measurements to Identify Stale Traceroutes. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. ACM Press, New York, NY, USA, 247–265. https://doi.org/10.1145/3419394.3423654

[20] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and kc claffy. 2014. Inferring Complex AS Relationships. In *Proceedings of the ACM Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. ACM Press, New York, NY, USA, 23–30. https://doi.org/10.1145/2663716.2663743

[21] Vasileios Giotsas, Georgios Smaragdakis, Christoph Dietzel, Philipp Richter, Anja Feldmann, and Arthur Berger. 2017. Inferring BGP Blackholing Activity in the Internet. In *Proceedings of the ACM Internet Measurement Conference* (London, United Kingdom) *(IMC '17)*. ACM Press, New York, NY, USA, 1–14. https://doi.org/10.1145/3131365.3131379

[22] Vasileios Giotsas, Georgios Smaragdakis, Bradley Huffaker, Matthew Luckie, and kc claffy. 2015. Mapping Peering Interconnections to a Facility. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies* (Heidelberg, Germany) *(CoNEXT '15)*. ACM Press, New York, NY, USA, Article 37, 13 pages. https://doi.org/10.1145/2716281.2836122

[23] Vasileios Giotsas, Shi Zhou, Matthew Luckie, and kc claffy. 2013. Inferring Multilateral Peering. In *Proceedings of the 9th ACM conference on Emerging Networking Experiments and Technologies* (Santa Barbara, California, USA) *(CoNEXT '13)*. ACM Press, New York, NY, USA, 247–258. https://doi.org/10.1145/2535372.2535390

[24] Enrico Gregori, Alessandro Improta, Luciano Lenzini, Lorenzo Rossi, and Luca Sani. 2011. BGP and Inter-AS Economic Relationships. In *International Conference on Research in Networking*. Springer, Springer, Valencia, Spain, 54–67.

[25] Enrico Gregori, Alessandro Improta, and Luca Sani. 2016. Isolario: A Do-ut-des Approach to Improve the Appeal of BGP Route Collecting. (2016).

[26] J Heitz, J Snijders, K Patel, I Bagdonas, and N Hilliard. 2017. RFC8092: BGP Large Communities Attribute. https://www.rfc-editor.org/rfc/rfc8092.txt

[27] Robert V Hogg, Joseph McKean, and Allen T Craig. 2005. *Introduction to Mathematical Statistics*. Pearson Education, Upper Saddle River, N.J.

[28] Packet Clearing House. 2005. Packet Clearing House.

[29] Geoff Huston. 2004. NOPEER Community for Border Gateway Protocol (BGP) Route Scope Control. RFC 3765. https://doi.org/10.17487/RFC3765

[30] Yuchen Jin, Colin Scott, Amogh Dhamdhere, Vasileios Giotsas, Arvind Krishnamurthy, and Scott Shenker. 2019. Stable and Practical AS Relationship Inference with ProbLink. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. USENIX Association, Boston, MA, 581–598. https://www.usenix.org/conference/nsdi19/presentation/jin

[31] The Wall Street Journal. 2021. Connecting the Fiber Glut: Level 3 to Buy Rival Global Crossing in Stock Deal Valued at 1.9 Billion. https://www.wsj.com/articles/SB10001424052748704529204576256541491117496 [Online; accessed 11-Oct-2021].

[32] Brivaldo A. Silva Junior, Paulo Mol, Osvaldo Fonseca, Ítalo Cunha, Ronaldo A. Ferreira, and Ethan Katz-Bassett. 2021. BGP Communities – Supplemental Material. https://github.com/TopoMapping/bgp-communities

[33] Richard M. Karp. 1972. *Reducibility among Combinatorial Problems*. Springer US, Boston, MA, 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9

[34] Ethan Katz-Bassett, Colin Scott, David R. Choffnes, Ítalo Cunha, Vytautas Valancius, Nick Feamster, Harsha V. Madhyastha, Thomas Anderson, and Arvind Krishnamurthy. 2012. LIFEGUARD: Practical Repair of Persistent Route Failures. In *Proceedings of the 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '12)*. ACM Press, New York, NY, USA, 395–406. https://doi.org/10.1145/2377677.2377756

[35] Thomas King, Christoph Dietzel, Job Snijders, Gert Doering, and Greg Hankins. 2016. BLACKHOLE Community. RFC 7999. https://doi.org/10.17487/RFC7999

[36] Maria Konte, Roberto Perdisci, and Nick Feamster. 2015. ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. ACM Press, New York, NY, USA, 625–638. https://doi.org/10.1145/2785956.2787494

[37] Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. 2020. Keep Your Communities Clean: Exploring the Routing Message Impact of BGP Communities. In *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies* (Barcelona, Spain). ACM Press, New York, NY, USA, 443–450. https://doi.org/10.1145/3386367.3432731

[38] Thomas Krenc, Robert Beverly, and Georgios Smaragdakis. 2021. AS-Level BGP Community Usage Classification. In *Proceedings of the 21st ACM Internet Measurement Conference* (Virtual Event) *(IMC '21)*. Association for Computing Machinery, New York, NY, USA, 577–592. https://doi.org/10.1145/3487552.3487865

[39] Tony Li, Ravi Chandra, and Paul S. Traina. 1996. BGP Communities Attribute. RFC 1997. https://doi.org/10.17487/RFC1997

[40] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2018. Internet Anycast: Performance, Problems, & Potential. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. ACM Press, Budapest, Hungary, 59–73. https://doi.org/10.1145/3230543.3230547

[41] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and kc claffy. 2013. AS Relationships, Customer Cones, and Validation. In *Proceedings of the 2013 Conference on Internet Measurement Conference* (Barcelona, Spain) *(IMC '13)*. ACM Press, New York, NY, USA, 243–256. https://doi.org/10.1145/2504730.2504735

[42] David Meyer. 1997. University of Oregon Route Views Archive Project.

[43] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O Twardziok, Alexander Kanitz, et al. 2021. Sustainable Data Analysis with Snakemake. *F1000Research* 10 (2021), 10–33.
[44] Bruno Quoitin and Olivier Bonaventure. 2002. *A Survey of the Utilization of the BGP Community Attribute.* Internet-Draft draft-quoitin-bgp-comm-survey-00. Internet Engineering Task Force. Work in Progress.
[45] Bruno Quoitin, Steve Uhlig, and Olivier Bonaventure. 2002. Using Redistribution Communities for Interdomain Traffic Engineering. In *Proceedings of the 3rd International Conference on Quality of Future Internet Services and Internet Charging and QoS Technologies 2nd International Conference on From QoS Provisioning to QoS Charging* (Zurich, Switzerland) *(QofIS'02/ICQT'02)*. Springer-Verlag, Berlin, Heidelberg, 125–134.
[46] Yakov Rekhter. 2006. RFC 4271: A Border Gateway Protocol 4 (BGP-4). https://www.rfc-editor.org/rfc/rfc4271
[47] NCC RIPE. 2021. RIPE RIS Project.
[48] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush. 2011. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. *IEEE Journal on Selected Areas in Communications* 29, 9 (Oct. 2011), 1810–1821. https://doi.org/10.1109/JSAC.2011.111006
[49] Johann Schlamp, Ralph Holz, Quentin Jacquemart, Georg Carle, and Ernst W. Biersack. 2016. HEAP: Reliable Assessment of BGP Hijacking Attacks. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1849–1861. https://doi.org/10.1109/JSAC.2016.2558978
[50] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V. Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. 2017. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. ACM Press, New York, NY, USA, 418–431. https://doi.org/10.1145/3098822.3098853
[51] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. 2018. ARTEMIS: Neutralizing BGP Hijacking within a Minute. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2471–2486. https://doi.org/10.1109/TNET.2018.2869798
[52] Petr Slavík. 1996. A Tight Analysis of the Greedy Algorithm for Set Cover. In *ACM STOC (STOC '96)*. ACM, Philadelphia, Pennsylvania, USA, 435–441.
[53] One Step. 2021. AS3257 Public Information. https://onestep.net/communities/as3257/. [Online; accessed 11-Oct-2021].
[54] One Step. 2021. AS3491 Public Information. https://onestep.net/communities/as3491/. [Online; accessed 11-Oct-2021].
[55] One Step. 2022. One Step. https://onestep.net/communities/
[56] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP Communities: Even More Worms in the Routing Can. In *Proceedings of the Internet Measurement Conference 2018* (Boston, MA, USA) *(IMC '18)*. ACM, New York, NY, USA, 279–292. https://doi.org/10.1145/3278532.3278557
[57] Alaa Tharwat. 2020. Classification Assessment Methods. *Applied Computing and Informatics* 17, 1 (2020), 168–192.
[58] Internet Tools. 2021. Whois Servers List. https://www.mobilefish.com/tutorials/whois_servers_list/whois_servers_list.html
[59] Feng Wang and Lixin Gao. 2007. On Inferring and Characterizing Internet Routing Policies. *Journal of Communications and Networks* 9, 4 (2007), 350–355. https://doi.org/10.1109/JCN.2007.6182869
[60] The Free Encyclopedia Wikipedia. 2021. Tier 1 Network. https://en.wikipedia.org/wiki/Tier_1_network [Online; accessed 11-Oct-2021].
[61] The Free Encyclopedia Wikipedia. 2021. Tier 2 Network. https://en.wikipedia.org/wiki/Tier_2_network [Online; accessed 11-Oct-2021].
[62] Jianhong Xia and Lixin Gao. 2004. On the Evaluation of AS Relationship Inferences [Internet Reachability/Traffic Flow Applications]. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, Vol. 3. IEEE, Dallas, TX, USA, 1373–1377 Vol.3. https://doi.org/10.1109/GLOCOM.2004.1378209
[63] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, and Ankur Jain. 2017. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. ACM Press, New York, NY, USA, 432–445. https://doi.org/10.1145/3098822.3098854
[64] Ying Zhang and Mallik Tatipamula. 2011. Characterization and Design of Effective BGP AS-path Prepending. In *2011 19th IEEE International Conference on Network Protocols*. IEEE, Dallas, TX, USA, 59–68. https://doi.org/10.1109/ICNP.2011.6089082

Received October 2021; revised December 2021; accepted January 2022