Automatic Inference of BGP Community Semantics

Brivaldo Alves da Silva Junior

Thesis

Subject Area: Computer Networks

Advisor: Prof. Ronaldo Alves Ferreira, Ph.D.

Co-advisor: Prof. Ítalo Fernando Scotá Cunha, Ph.D.

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul
September, 2024

Automatic Inference of BGP Community Semantics

Brivaldo Alves da Silva Junior

Thesis

Subject Area: Computer Networks

Advisor: Ronaldo Ferreira, Ph.D.

Co-advisor: Ítalo Fernando Scotá Cunha, Ph.D.

Submitted in partial fulfilment of the requirements for the degree of Doctor in Computer Science

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul
September, 2024

Acknowledgments

The construction of this thesis is the culmination of a long, hard, and very time-consuming work that represents, in many ways, the result of a learning process. None of this would be possible without the support of countless people and God on this journey. I thank my beloved wife, Elizângela; my children Lucas, Diogo, and Beatriz; my parents, Brivaldo and Nina Rose; my brothers Bruno and Breno; my mother-in-law Marli; my father-in-law Waldir (in memoriam); my advisor, Professor Ronaldo, and co-advisor, Professor Ítalo; and the most diverse friends on the journey who contributed directly or indirectly with me.

Over this long journey, I had the pleasure of having a dedicated advisor. During this time, I had life experiences that I would hardly have had with another advisor at UFMS. In 2018, I had the honor of having my first international work, a poster, accepted to the ACM SIGCOMM. I was fortunate to present it in person in Budapest. In 2022, I had another paper accepted at the ACM SIGMETRICS, which I presented remotely. I also had the opportunity to interact in person with very high-level international researchers, such as Professor Jennifer Rexford from Princeton, Doctor Walter Willinger from NIKSUN, Inc., and Professor Ethan Katz-Basset from Columbia. I am very grateful to Professor Ronaldo for all the guidance and support throughout this journey.

I am also very grateful to my co-advisor, Professor Italo, who is a wonderful person. I had the opportunity to interact with his students at UFMG, such as Osvaldo, Paulo, and Elverton, brilliant students, and collaborate with some of Osvaldo's work that allowed me a greater understanding of the environment I was entering into to carry out the research for this thesis. I am equally grateful to Professor Italo for all his guidance and for agreeing to participate in this journey.

It is difficult to name all the people who helped me throughout this process, but some of them made great contributions in different aspects and moments. I name them in lexicographic order: Adriano, Airton, Alexandre, Diego, Fabricio, Paulo, Samuel, Tiago, and Victor, wonderful people who contributed to this work becoming a reality.

I would also like to thank professors Edmundo Albuquerque de Souza e Silva (UFRJ), Marinho Barcellos (University of Waikato, New Zealand), Pedro Marcos (FURG) and Carlos Alberto da Silva (UFMS), for being part of my doctoral thesis defense committee and for their valuable and constructive comments to improving this work.

Finally, I would like to thank the financial support of the Federal University of Mato Grosso do Sul (UFMS) and RNP, the Brazilian National Research and Education Network (NREN). UFMS and the College of Computing (Facom) granted me a leave of absence to dedicate myself to the doctoral program, which allowed me to have this academic experience. RNP funded a fellowship for me to work on the research project "Researching Internet Routing Security in the Wild".

Abstract

The Border Gateway Protocol (BGP) orchestrates Internet communications between Autonomous Systems (ASes). BGP's flexibility allows operators to express complex policies and deploy advanced traffic engineering systems. A key mechanism for 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.

The main goal of this thesis is to develop techniques to infer the semantics of BGP communities using publicly available data from BGP collectors. We first propose a set of techniques to infer location communities. 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 (93%), better coverage (81% recall), and dynamic updates, complementing operators' and researchers' abilities to reason about BGP community semantics.

We also design and evaluate algorithms to automatically uncover BGP *action communities* and ASes that violate standard practices by consistently using the *informational communities* of other ASes, revealing undocumented relationships between them (*e.g.,* siblings). Our experimental evaluation with billions of route announcements from public BGP route collectors from 2018 to 2023 uncovers previously unknown AS relationships and shows that our algorithm to identify action communities achieves average precision and recall of 92.5% and 86.5%, respectively.

Keywords: Internet routing, BGP Communities, AS Relationships

Contents

List of Acronyms

ACM	Association for Computing Machinery
ARIN	American Registry for Internet Numbers
AS	Autonomous System
ASN	Autonomous System Number
AS-to-Org	Autonomous System-to-Organization
BGP	Border Gateway Protocol
c2p	Customer-to-Provider
CAIDA	Center of Applied Internet Data Analysis
CDN	Content Delivery Network
DDoS	Distributed Denial-of-Service
GT	Ground Truth
GTT	Greater Technology Together
IETF	Internet Engineering Task Force
IFIP	International Federation for Information Processing
IP	Internet Protocol
IRR	Internet Routing Registry
ISP	Internet Service Provider
IXP	Internet Exchange Points
LACNIC	Regional Internet Registry for the Latin American and Caribbean
MED	Multi-Exit Discriminator
MRT	Multi-Threaded Routing Toolkit
NCC	Network Coordination Centre

NLP	Natural Language Processing
NREN	National Research and Education Network
p2c	Provider-to-Customer
p2p	Peer-to-Peer
PCH	Packet Clearing House
PeeringDB	Database of Internet Peering Networks
PoP	Point of Presence
REN	Research and Education Networks
RFC	Request for Comments
RIB	Routing Information Base
RIPE	Réseaux IP Européens
RIS	Routing Information Service
RV	Route Views
VP	Vantage Point

List of Figures

List of Tables

Chapter 1

Introduction

> *Some things in life can never be fully appreciated nor understood unless experienced firsthand. Some things in networking can never be fully understood by someone who neither builds commercial networking equipment nor runs an operational network.*
>
> – RFC 1925, *The Twelve Networking Truths*

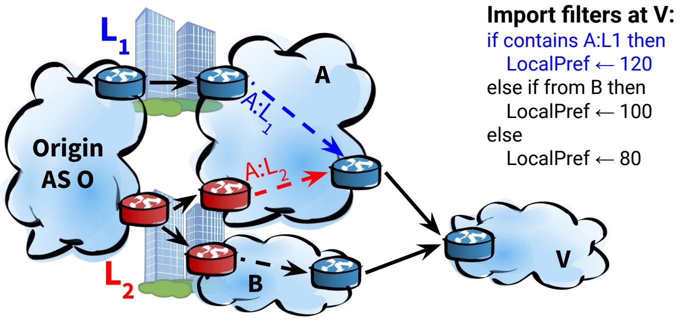
The Internet is a complex system composed of Autonomous Systems (ASes) that exchange reachability information using the Border Gateway Protocol (BGP) [79,80, 82], its *de facto* interdomain routing protocol. The construction of a route in BGP starts from an *origin AS*, which controls and announces an IP prefix to its neighboring ASes over BGP. Routes are propagated by BGP *updates*, routing messages composed of mandatory and optional attributes. Mandatory attributes include, *e.g.,* the destination IP prefix announced by the origin, the next-hop router's IP address, and the AS-path. The *AS-path* is the sequence of ASes traversed by the route until it reaches the origin AS. The AS-path is employed by BGP to prevent loops, and more rarely by operators to specify complex routing policies [34]. Optional attributes can be transitive or non-transitive depending on whether they are carried over by a neighboring AS to the next AS and include *communities* and *multi-exit* *discriminators.*

The BGP best-path selection algorithm is flexible and allows network operators to rank routes based on policies and economic agreements. BGP offers various parameters to control routing decisions, such as setting route preferences (LocalPref), signaling preferred interconnections between neighboring ASes (MEDs), and minimizing intradomain traffic costs. However, these mechanisms are coarse-grained and primarily control decisions for routes received from neighboring ASes. The increasing demand for reliability and performance has led to more dynamic and complex routing policies [37, 85, 93, 105], exposing the limitations of a protocol that was last updated more than two decades ago [79].

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. A BGP community is a 32-bit tag whose meaning (*i.e.,* semantics) is defined independently by each network. However, network operators generally group BGP communities into two types: *informational* and *action.* A network tags a route with an *informational community*[1] to convey information to its neighbors, such as the country, city, PoP, or router where it learned the route [36, 63] or the business relationship with the neighboring network from where it received the route [37, 50, 64]. On the other hand, a network tags a route with an *action community* to request an action from an upstream network [17, 93, 108]. An action community can request the network to prepend its AS number to the BGP AS path to make a route artificially longer and less attractive, or to not advertise a particular prefix to one of the newtork's peers to steer traffic destined to that prefix away from a low-performance AS.

BGP communities provide various options for traffic engineering. For instance, Figure 1.1 illustrates an example where AS V uses a location community, which is a type of informational community, to control route selection. The origin AS O

[1] In this thesis, we use the term *informational community* interchangeably with *information community.*



```
Import filters at V:
if contains A:L1 then
   LocalPref ← 120
else if from B then
   LocalPref ← 100
else
   LocalPref ← 80
```

Figure 1.1: 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.

announces its 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 performances 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 [99] or webpages (*e.g.,* [32]), but most of the communities observed in public route announcements are undocumented.

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 rely on ad-hoc information in IRR databases or webpages, which may be incomplete, outdated, or available only by contacting 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.

1.1 Problem Statement and Research Questions

A recent study introduces a mining tool designed to automatically build a database of BGP community semantics by extracting information from IRR records and support webpages of network providers [35]. The tool uses natural language processing to infer the meaning of each documented community within these data sources. Although the study [35] shows that the tool achieves high precision in identifying communities, its approach has two fundamental limitations: *(i)* it can only infer a restricted number of communities, as it depends on free text descriptions provided by network operators; and *(ii)* it is constrained by the quality of data sources, which may be incomplete, outdated, or entirely missing, leading to reduced precision and limited coverage of communities used across the Internet. Additionally, an AS can use the BGP communities of a related AS, such as a sibling AS. This practice complicates the understanding of BGP community usage, as an informational community may appear in a route announcement even when the AS that originally defined it is not present in the announcement. Consequently, network operators still have to rely on manually created documentation provided by each individual AS about their BGP communities and relationships with other ASes, which is often incomplete and insufficient for effective troubleshooting and understanding of Internet routing. Thus, our problem statement can be summarized as follows:

Problem Statement 1: *Networks do not publicly provide necessary and sufficient information about their BGP communities and relationships with other networks for effective troubleshooting and understanding of Internet routing.*

Many studies have identified the lack of data documentation as a major problem in troubleshooting Internet routing issues [22, 25, 35, 36, 60, 75, 93]. In this thesis, we contribute to partially close this gap by automatically building reliable databases to document a subset of communities that are actively being used on the Internet, *i.e.,* communities that appear in public BGP route collectors. We also present mechanisms to uncover an undocumented type of confounding use of BGP communities in the wild in which an AS consistently uses the informational communities of another AS, which might help operators understand BGP community uses or uncover undocumented AS relationships. This behavior can impact previous research that infers AS relationships or the semantics of BGP communities [37, 59, 60, 64, 89, 93]. More specifically, we focus on the following two research questions:

Research Question 1: *Can we build reliable databases of BGP community semantics using public routing data?*

In this thesis, we address this research question by developing techniques to automatically infer the semantics of specific types of BGP communities directly from publicly available route announcements collected by route collectors. We initially target *location communities* (Chapter 4), defined as communities that carry metadata about the location (*e.g.,* city, country, continent, router, PoP, link, or interconnection) where a route was learned. 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 initially on location communities because they represent the majority of publicly-documented communities (§4.2) as well as a significant fraction of communities observed in route announcements (§4.3). 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 (*e.g.,* latency, jitter, etc.) to tune their route selection preferences at a finer granularity than possible with just AS paths, as exemplified by Figure 1.1.

In Chapter 5, we also present algorithms for identifying action communities from public routing data. Recall that an action community is a tag that an AS inserts in a route announcement to request an action from an upstream network. Our algorithms provide automatically updated metadata (*i.e.,* a database of action communities) that can benefit novel tools and models. For example, action community information can help operators troubleshoot routing anomalies, *e.g.,* when routes that follow an unexpected or undesired path carry specific action communities, and identify opportunities for traffic engineering, *e.g.,* when an operator observes preferable routes induced by action communities not publicly documented. Our results can also be used to help identify and flag announcements carrying BGP communities to perform route manipulation attacks [4, 5, 70].

In Chapters 4 and 5, we show through longitudinal studies that our algorithms perform well over the years even when ASes add new communities or decomission old ones, attesting to their robustness over time and the reliability of the generated databases.

Research Question 2: *Can we use BGP communities to identify AS relationships?*

The main goal of this thesis is to determine the semantics of BGP communities. However, in our analysis of routing announcements, we observed that, although rare, different ASes can use each other's communities even when they are not siblings. Operationally, this behavior is more commonly expected from sibling ASes, as the same organization manages them. Yet, during our inference process, we found that nonsibling ASes also exhibited this behavior, complicating the inference of location communities. This complication arises because the location community appears in route announcements without its corresponding AS being present in the AS path, which deviates from the expected behavior.

In Chapter 4, we build a heuristic based on the hitting set algorithm [31] (equivalent to the NP-complete vertex cover problem [31, 54]) to detect the presence of these ASes that use the communities of others and prevent that their presence excludes location communities from the inference. As we deepened our understanding of community usage, we discovered that these relationships were not limited to sibling ASes, as some ASes use the communities of other ASes even when they are not siblings. We call this behavior *community squatting*[2] and identify the ASes involved as *AS squatters*.

In Chapter 5, we use BGP communities to identify squatting relationships and reduce the noise they introduce into the inference of action communities. Informational communities are helpful for this task because they are expected to appear consistently with their respective ASes (or related ASes) in the AS path. However, our algorithm does not need to know the type of a community to infer the squatting relationships.

Although there is no ground-truth dataset on AS squatting relationships, we

[2] We borrow the term *squat* and its derived forms from "IP address squatting" [83], where a network uses another's IP address space internally for its own purposes. In this work, however, an AS may *squat* the communities of another AS legitimately, *e.g.*, the communities of a sibling AS.

were able to compare our inferred AS relationships with techniques that use public data and databases such as PeeringDB [3]. Our inference mechanism captured relationships *in the wild* that the existing techniques missed, thus addressing Research Question 2. Additionally, our algorithm to uncover squatting relationships can complement techniques for validating AS-relationship inferences and tracking route changes.

1.2 Main Contributions

In this section, we present our main contributions to the automatic identification of location (a sub set of the information communities) and action communities on the Internet. We treat these sets separately because they require different techniques and use BGP dumps from different time periods. For location communities, we analyze data from 2017 to 2020, while for action communities, we analyze data from 2018 to 2023. Chapters 4 and 5 detail the algorithms and techniques developed in this work.

We also design algorithms to identify ASes that engage in uncommon practices by consistently *squatting* on the BGP communities of other ASes, which we refer to as a *squatting relationship.* This behavior can affect the validity of previous research that relies on BGP communities [37, 59, 60, 64, 89, 93].

1.2.1 Location Communities

Our approach to infer location communities is fundamentally different from previous efforts, as our algorithms automatically infer communities from public route announcements observed by BGP route collectors (*e.g.,* RouteViews [68], RIPE RIS [81], and Isolario [42]) and generate databases of communities that can be regenerated any time to reflect additions of new communities or assignment changes. *Our work is the first to show that we can use routing announcements to infer, even at a coarse level, the semantics of BGP communities.* Our key insight for inferring location communities 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 [42, 68, 81] and infer 15,505 location communities across 1,120 ASes, wich represents 19.67% of the communities that appeared in the BGP dumps in 2020.

We evaluate our inference methodology for identifying location communities 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 (from 87% to 93%) and recall (from 72% to 81%) depending on the parameters used. We compare our results with CAIDA's manually-built public database of BGP communities [7] and show that our technique 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 [53].

1.2.2 Action Communities and AS Squatters

Our approach to infer action communities relies on public route announcements observed by BGP route collectors (*e.g.,* RouteViews [68] and RIPE RIS [81]). It fundamentally differs from previous efforts that rely on public documentation about BGP communities—published by the networks on Internet Routing Registries (IRRs) or web pages—as a basis for classifying undocumented communities [60] or to extract community semantics using natural language processing [35, 38]. As discussed previously, these approaches do not generalize well to ASes that do not follow common practices to define their communities or are limited in the number of communities they can infer because they depend on free text descriptions provided by network operators, which may be incomplete or outdated.

Our key insight lies in the fundamental difference between the usage of informational and action communities. *Informational communities* are used by ASes to pass information to other ASes, such as where the AS learned a route or its business relationship (customer, provider, or peer) with the previous AS on the route. Consequently, an informational community should appear on routes that traverse the community's AS, as the AS is in charge of tagging routes with the relevant information. In contrast, an action community is less likely to be tagged on the routes where its AS is present, as the community sends a request to and is tagged by a network other than the AS that defines the community, which we refer to as *controlling AS*. Furthermore, RFC7454 prescribes that the controlling AS remove an action community from a route after performing the requested action [23]. Therefore, if the AS that defines the community is on the route, it should have removed its action communities. As such, an action community should only appear if the route does not traverse its AS. Our algorithms rely on this fundamental difference to build reliable classifiers of action communities and to identify *potential squatters*.

While our insight is simple to state, designing algorithms that perform well in the wild presents significant challenges, such as ASes that squat the information communities of other ASes, ASes that do not remove their action communities after performing the requested actions, route announcements with a large number of communities, and limited visibility of the existing BGP route collectors. We address these challenges by identifying squatting relationships based on how routes with informational communities propagate on the Internet. Then, we build an initial set with action communities that are mostly absent from routes traversing the ASes that define them. Using this initial set, we construct an efficient data structure to identify action communities in route announcements where the ASes that define them can be present in the AS-paths. Although we cannot control how manufacturers set the default behavior on their devices, it is possible that removing communities from the BGP announcements is related to security concerns. However, this claim requires further investigation and is outside the scope of our current work. We contacted multiple network operators to understand their reasons for removing or not removing action communities to no avail.

Our experimental evaluation with billions of route announcements from 2018 to 2023 shows that the algorithm to identify action communities achieves average precision and recall of 92.5% and 86.5%, respectively, over all communities in BGP dumps covered by our ground truth in the longitudinal study. We also analyzed over 739 million announcements from December 2023 and inferred 19,564 action communities from 2,099 autonomous systems. We excluded 14.86% of the communities (15,800 out of 106,262) from our evaluation in 2023 due to their association with private ASNs. Our algorithm for uncovering potential squatters found 54 pairwise squatting relationships involving 105 ASes that systematically used other AS's communities in December 2023. These identified squatting relationships may uncover undocumented relationships between the ASes. For example, we identified five sibling relationships that the state-of-the-art technique described in [12] did not detect.

Our algorithms provide automatically updated metadata (*i.e.,* a database of action communities and potential squatters) that can benefit novel tools and models. For example, action community information can help operators troubleshoot routing anomalies, *e.g.,* when routes that follow an unexpected or undesired path carry specific action communities, and identify opportunities for traffic engineering, *e.g.,* when an operator observes preferable routes induced by action communities not publicly documented. Our results show that operators use action communities much more extensively than publicly available documentation would indicate.

1.3 Thesis Roadmap

The remainder of this thesis is organized as follows. Chapter 2 presents an overview of the BGP protocol and introduces concepts and terms used in the subsequent chapters. In Chapter 3, we review related work, including recent efforts to characterize community usage, standardization initiatives, methods to infer community semantics, and various applications of BGP communities. Chapter 4 outlines our techniques for inferring location communities, discussing the underlying assumptions, datasets, and the performance of these techniques. Chapter 5 presents our methods for detecting AS squatters and identifying BGP action communities, together with a discussion of our assumptions, datasets, and evaluation results. Finally, Chapter 6 presents the thesis's conclusions and suggests directions for future research.

Chapter 2

Background

> *"Begin at the beginning," the King said, gravely, "and go on till you come to an end; then stop."*
>
> – Lewis Carroll, *Alice in Wonderland*

This chapter introduces the key characteristics of the Border Gateway Protocol (BGP), the role of commercial relationships in traffic exchange between Autonomous Systems (ASes), and the impact of certain optional BGP attributes on routing decisions. These concepts are crucial to understanding the techniques discussed in this thesis.

2.1 BGP Protocol

BGP is the *de facto* interdomain routing protocol on the Internet, used to exchange reachability information across ASes. It combines *scalability* to large numbers of prefixes and routes with *flexibility* to support complex routing policies and protect business secrets. The proof of BGP's Turing completeness [15] attests to its flexibility.

BGP's flexibility is manifested by its *best-path selection algorithm*, which AS routers use to choose one among multiple routes available towards a prefix. The BGP path selection algorithm is a multi-stage decision process that starts by comparing the local preference attribute (LocalPref) of known routes. The LocalPref is an operator-defined integer used to rank routes in a way that captures business interests. Most commonly, LocalPref is set to implement policies that follow the Gao-Rexford model [30] (Section 2.2), where an AS prefers income-generating routes received from *customers*, then routes received from settlement-free *peers*, and finally cost-incurring routes from transit *providers*. As a result, the most important property in choosing routes in BGP is performance-oblivious, as discussed in previous works [63, 85, 94, 105].

The second stage in the BGP path selection algorithm compares the length of the AS-paths of the available routes, with the shortest AS-path being preferred. A common practice in the Internet is BGP AS-path prepending [17, 65, 76, 107, 108], where an AS purposefully prepends its AS number multiple times to the AS-path, making it artificially longer and thus less preferred.

The BGP path selection algorithm proceeds with a sequence of tie-breakers in case multiple equally-preferred routes of equal length are available. Notable among them are the comparison of Multi-Exit Discriminators (MED) [79] of multiple routes received from one neighboring AS over distinct links, and the comparison of the cost metric of the intradomain routing protocol. These two stages consider intradomain routing and control which link is used between two ASes to exchange traffic. Common policies implemented through MEDs and intradomain costs include hot- and cold-potato routing, which reflect whether an AS tries to shed traffic to other ASes as soon as possible (*e.g.,* to minimize cost) or is willing to carry the traffic towards the destination as far as it can (*e.g.,* to improve performance) [24,96]. The algorithm concludes with tie-breakers that introduce randomness: prefer the oldest route (*i.e.,* the one received first), prefer the route received from the router with the lowest identifier, or simply choosing a route at random.

2.2 AS Relationships

The Gao-Rexford model [30] 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 (ISP), 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 [29]. 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 [20, 40, 64].

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. In the Gao-Rexford model [29], 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.

2.2.1 Valley-free Routing

As customer networks pay providers for transit, money flows up in the AS hierarchy. As a result, Internet routes usually traverse a sequence of customer-to-provider links, zero or one peer-to-peer link, and a sequence of provider-to-customer links [29, 30]. Routes with this property are called *valley-free*, and often described as a hill: The *uphill* region starts at the origin AS and includes consecutive AS-pairs with customer-to-provider links; it captures the region of the path when traffic is going up the AS hierarchy and traveling towards the "core" of the Internet. The *peak* is either the single AS at the top of the hill when no peer-to-peer link exists, or the AS-pair at the top of the hill if a peer-to-peer link exists. The *downhill* region includes the remaining consecutive AS-pairs with provider-to-customer links up to the last AS; it captures the region of the path when traffic is going down the AS hierarchy, away from the "core".

Valley-free routing has positive implications as it implies BGP converges to a stable solution [30]. Violations of valley-free routing can be caused by complex AS relationships [37], *e.g.,* partial transit, sibling ASes, Research and Education Networks (REN) exporting routes from one peering REN to another peering REN, and misconfigurations like route leaks [93, 106] or prefix hijacks [97, 106]).

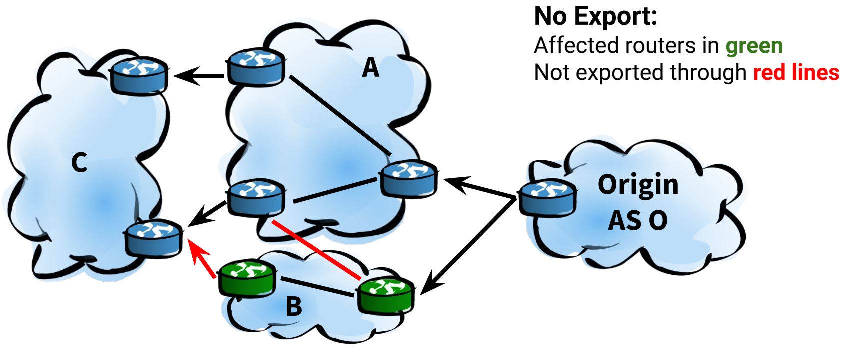


Figure 2.1: ASes *A* and *B* distribute the routes received from Origin AS O tagged with no-export locally but do not export them to their neighboring ASes.

2.3 BGP Communities

The use of BGP communities has increased significantly in the past few years (§5.5, [93]). 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 [77], but operators have not fully embraced these proposals. Only a handful of community values have been standardized [46, 56, 62]. For example, 65535:666 (blackhole) signals a request to an upstream network that traffic to a destination prefix should be dropped [56], 65535:65281 (no-export) signals that the prefix should not be exported outside the AS (Figure 2.1), and 65535:65284 (no-advertise) signals a request to a provider that a route should not be advertised further [62] (Figure 2.2). Standardized 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 [91]. The lack of documentation on communities and the ad-hoc nature of available documentation constrains our understanding of Internet routing.

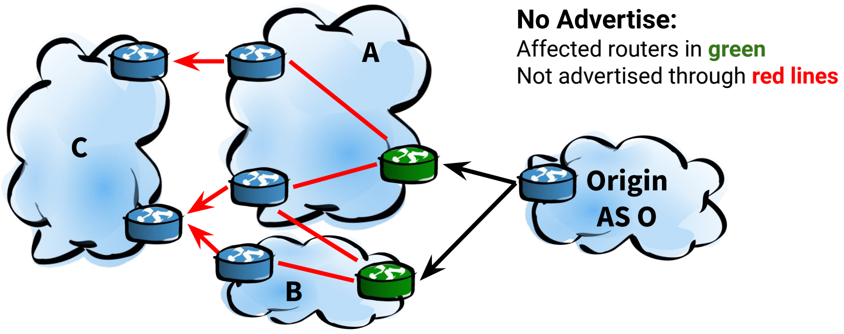


Figure 2.2: BGP border routers that receive an announcement with community no-advertise learn the route but do not forward the announcement even inside their ASes.

Despite the flexibility built into the BGP best path selection algorithm, BGP communities provide additional flexibility to support, *e.g.,* more complex or fine-grained traffic engineering policies [62]. A BGP community is a 32-bit tag[1] that can be attached to an announcement, and an announcement can carry an arbitrary number of BGP communities. They are an optional transitive attribute, meaning BGP communities should propagate broadly. However, all router vendors provide configuration options to remove communities from the announcements, and some vendors remove communities by default, possibly due to security concerns (*e.g.,* Cisco [47]). This limits community propagation and visibility in an uncontrolled manner [59], which imposes a challenge to inferring BGP community semantics and to our work.

Network operators have flexibility in defining semantics for BGP communities,[2] and are limited only by the (increasing) community-handling capabilities of BGP routers. The semantics of the BGP community fits into two classes [22, 58, 59, 67]:

- *Informational communities* add metadata to a route announcement. Use cases include assisting operators with traffic engineering [63], troubleshooting issues, refining policies, and capacity planning [36, 64, 104]. Example metadata in informational communities include specifying whether a route was originated either internally or learned externally; whether external routes were learned from a customer, provider, or peer; or the location (city, country, or region) where the route was learned or originated. Informational communities may be used by the controlling AS itself as well as downstream ASes.

  In Chapter 4, we infer *location communities*, which are information communities that indicate where a route was learned. 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 indicates a geographical location.

- *Action communities* signal an action that an AS executes on behalf of another AS and are usually used to trigger actions at a provider on behalf of a customer. Action communities generally influence the BGP path selection process or how routing announcements propagate to realize some traffic engineering policy [5, 59, 67]. Examples include adjusting LocalPref to make the route less preferable at the transit provider, prepending the BGP AS-path to make it longer and thus less preferable for other ASes (often used for backup routes), and constraining route propagation to a subset (or none) of the transit

[1] We consider only 32-bit communities [62], which work better for ASes with 2-byte AS numbers. BGP *extended* communities [95] and BGP *large* communities [43] are 64- and 96-bit long, respectively, and include support for 4-byte ASNs. Although we do not analyze extended or large communities in this work as their use is incipient, our techniques can be applied to them.

[2] Only 15 "well-known" communities have predefined semantics in IETF standards [48].

provider's neighbors. BGP communities can even impact traffic forwarding on the data plane, as it is commonly used to blackhole malicious traffic [38, 93].

In Chapter 5, we show how we identify action communities. These communities work to modify Internet routing announcements to improve or control the flow of traffic across the network. However, these communities can make traffic analysis challenging because they can tag links that are never visible to route collectors. To address this challenge, we investigate hidden relationships (*e.g., AS squatters*) and identify those that actively influence route announcements.

The standards suggest, and most operators (but not all) follow the convention that the first 16 bits represent the AS number (ASN, a number that identifies the AS) of the AS that defines the community's semantics, in this thesis referred to as the *controlling AS*, and that the last 16 bits is an arbitrary operator-defined value [62]. For example, 3356:70 is an action community that asks Level 3 (now Lumen Networks, AS3356) to decrease the LocalPref of a route to 70 (from the default 100), making the route less preferable; while 3356:2009 is an informational community added by Level 3 to routes learned at San Francisco.

In recent years, network operators have seen an increase in the adoption of BGP communities [22, 93]. The percentage of routes received by BGP collectors with at least one community increased from 59% in 2018 to 71% in 2023, even taking into account the 3,07× increase (from 161,878,003 to 496,846,470) in the number of BGP routes in public table dumps across all RouteViews and RIPE RIS collectors [68, 81]. There is also ongoing effort to increase the capabilities of BGP communities, such as the new proposal of arbitrary length *wide.* BGP communities under standardization by the IETF [78].

2.3.1 Route Collectors

Research on Internet routing requires access to routes distributed by the BGP protocol. Existing platforms, such as the University of Oregon Route Views (RV) [68], and

RIPE NCC Routing Information Service (RIS) [81], deploy tens of routers strategically positioned to collect BGP updates from hundreds of collaborating ASes, generating datasets with millions of records every day. These openly accessible datasets help researchers and network operators gain insights into the propagation of BGP announcements and reason about the dynamics of Internet routing. The routers of these projects are distributed across diverse physical locations worldwide and offer opportunities for observing how routes tagged with BGP communities propagate and how the ASes along a path deal with them. One AS may have multiple routers peering with BGP collectors located at different geographical locations, providing enhanced visibility to its routing policies. We denote each of these routers as a vantage point (VP).

2.4 Summary

This chapter discusses BGP, the protocol that coordinates Internet routing. BGP is designed with attributes that facilitate the distribution of prefix reachability information across the Internet. The complex relationships between ASes on the Internet are implemented by carefully crafting the information that each BGP announcement should carry.

Operators can use the BGP attributes of route announcements to collect information or influence protocol decisions, allowing them to partially control traffic flow to their ASes.

Chapter 3

Related Work

> *Every old idea will be proposed again with a different name and a different presentation, regardless of whether it works.*
>
> – RFC 1925, *The Twelve Networking Truths*

This chapter presents recent efforts to define, standardize, and understand the semantics of BGP communities, and their legitimate and malicious uses. We also discuss recent work to infer AS relationships, which is relevant for understanding routing policies and dealing with ASes that use communities of their siblings.

3.1 Characterization of Community Usage

The use of BGP communities has been explored in several ways. We first discuss some works that explore and characterize how operators use BGP communities.

Streibelt *et al.* [93] 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 previously believed, sometimes reaching ASes several hops away from the intended target of the community. Over half of the communities traverse more than four ASes, with 10% exceeding six. 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. Our work is one step in this direction, as it provides a database of community semantics that can be automatically and periodically updated.

Krenc *et al.* [59] propose an algorithm that uses only passive measurements, just like ours, to infer how ASes handle communities. BGP communities are a transitive attribute of BGP updates, which means that they should propagate from one AS to the next; however, routers can be configured to filter them. The proposed algorithm infers whether an AS forwards or discards communities from the BGP announcements.

3.2 Standardization efforts

Due to the increasing use of BGP communities, there have been several efforts to standardize their use or, at least, to classify how operators are using communities on the Internet.

Quotin and Bonaventure [75] identify two main uses for BGP communities on the Internet: provide information or instruct a network on how to handle a route. Informational communities are tagged on routes to, for example, specify the type of neighbor that exported a route or a location where a route was received. Action communities, on the other hand, instruct a neighbor or remote network on how to handle a route to, for example, perform traffic engineering.

Donnet and Bonaventure [22] extend the classification proposed in [75] and investigate the use of BGP communities in the wild. Their findings reveal a significant increase in the use of BGP communities, suggesting that network operators recognize their potential value for influencing routing behavior. However, the study also identifies a critical limitation: the lack of standardized definitions for these communities. The effectiveness of community use is hampered without understanding what the community values mean. This lack of definitions highlights the need for better practices and standardization of BGP community semantics to unlock their full potential for improving Internet routing control and optimization.

3.3 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 [35,38]. These data sources are generally incomplete and outdated, significantly limiting the number of communities that approaches based on NLP can achieve. These approaches report high precision, but their coverage is very limited, as only a small number of ASes document their communities in public repositories. On the other hand, our approach automatically generates an up-to-date database from BGP dumps that contains BGP communities currently in use by the network operators, increasing, therefore, coverage and precision. As the time to generate the database is not significant (just a few hours), we can regenerate the database as needed.

Krenc *et al.* [60] propose a clustering algorithm for classifying information and action communities that depends on a ground-truth database to define the parameters that separate the two types of clusters. The paper shows high precision for the algorithm. However, the evaluation uses the same communities that were used to define the parameters of the algorithm, *i.e.,* it doesn't split the communities into training and test datasets to determine if the parameters generalize to the test dataset. As we show in Section 5.5, their approach may not generalize to other ASes, resulting in lower precision and recall for the action communities than the ones reported in [60].

The approaches of Giotsas *et al.* [35, 38] and Krenc *et al.* [60] depend on the availability of documentation from the ASes, which is sometimes incomplete, outdated, or nonexistent. In our work, we use the existing documentation only to build the ground-truth database and evaluate the results of our inference algorithms. Our inference algorithm use public information from route collectors to infer location communities (Chapter 4), which are information communities that tell where a route is learned, and action communities (Chapter 5) used to manipulate, tuning and improve the route announcements on the Internet.

3.4 Legitimate Uses of BGP Communities

Determining the relationship between two ASes is a hard problem, but it has many important applications [64]. 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. While network operators can request community lists and semantics from their direct providers, challenges remain when they need semantic information for ASes that are not directly connected to their own networks. This issue is even more pronounced for passive observers, such as researchers, who observe routing announcements without actively participating in the routing process.

Feldman *et al.* [25] use BGP communities to monitor updates across observation points and prefixes to detect route instabilities. They observe changes in AS path, origin, MED, next-hop, and communities to determine a faulty set of links. The main idea is that changes in the best path for a prefix may indicate routing instabilities. We explain in Chapter 4 a method that can significantly improve the detection process. We built a dictionary of location communities that can be used to track changes in internal routes even if the AS path does not change.

Giotsas *et al.* [35] shows that a reliable dictionary of BGP communities can significantly improve the detection of infrastructure outages. They propose Kepler, a tool that uses public data sources (RIPE Atlas, Routeviews, PeeringDB, and

DataCenterMap) to correlate BGP information from BGPStream and PathCache to build a map of buildings and routing systems around the world to find outages. They discovered that the number of public outages reported is significantly smaller than the real number of problems in network infrastructure places. To create a map and a base of communities to find outages, Kepler first extracts tokens with the names of cities and airports from IRR (Internet Routing Registry), websites, and possible communities that determine location. Next, Kepler observes communities related to geolocation or facility location to infer Internet Exchange Points (IXP) and Network Facilities. Kepler uses traceroute to build the network topology with the paths that are considered normal. When something changes, Kepler runs traceroutes again and collects BGP data to observe modifications (AS-path or communities with local information).

The work in [36] aims to identify changes within a network domain (intradomain path changes) by looking for modifications in BGP communities. The goal is to maintain an up-to-date collection of traceroutes to help in traffic engineering, ultimately improving the performance of Content Delivery Networks (CDNs), for example. As using random traceroutes is expensive to detect changes in the Internet topology, Giotsas *et al.* [36] use BGP feeds to identify these changes and trigger reruns of traceroutes. However, correlating BGP feed changes with actual network topology modifications is challenging, as traceroute probes are not guaranteed to be located in the same places as the BGP collectors. To address this limitation, the authors propose using changes in BGP communities that might signal a switch in the border router, prompting new traceroutes. Our work can significantly improve this technique by providing a comprehensive dictionary of location and traffic engineering communities.

Li *et al.* [63] present a measurement study that reveals that anycast paths can suffer from path inflation, meaning that the chosen path to a replica server may not be the most efficient (shortest latency or fewest hops). Although anycast is widely used for critical network infrastructure due to its ability to provide one-to-any communication, its performance can fall short of expectations, as BGP routing lacks mechanisms to prioritize paths based on performance. For instance, path inflation occurs when BGP selects suboptimal routes with the same AS path length when presented with multiple announcements. The authors propose an improvement that encodes replica servers' geographic coordinates within BGP community tags. This encoding would allow BGP to select nearby replicas during the route selection process.

3.5 Malicious Uses of BGP Communities

Some works have shown that BGP communities can be a vector for malicious attacks [5,93]. Figure 3.1 illustrates an example using the blackhole community. Using BGP communities for this purpose is justified because intercept attacks based on prefix hijacking generally disrupt significant parts of the Internet [84], which induces rapid detection and remediation by network operators. SICO [5], on the other hand, builds community-based intercept attacks that target small parts of the Internet and are harder to detect. The attack focuses on hijacking a BGP prefix and intercepting its traffic, but ensuring the traffic flows between source and destination. The attacker makes a bogus announcement—*i.e.,* a path that includes its AS to reach the victim AS—to an upstream provider and a legitimate one to another provider. The attacker uses do-not-export communities to limit the propagation of the bogus announcement and intercept selected prefixes only. It also keeps a path to the actual origin to forward the intercepted traffic.

Streibelt *et al.* [93] present several scenarios in which a malicious actor can abuse BGP communities to launch several types of attack, as we mentioned above, such as remotely triggering blackholing, steering traffic away from its normal trajectory, and route manipulation. These attacks generally rely on action communities, such as the blackhole and no-export communities, and improperly configured routers that forward non-transitive communities. We infer location and action communities.

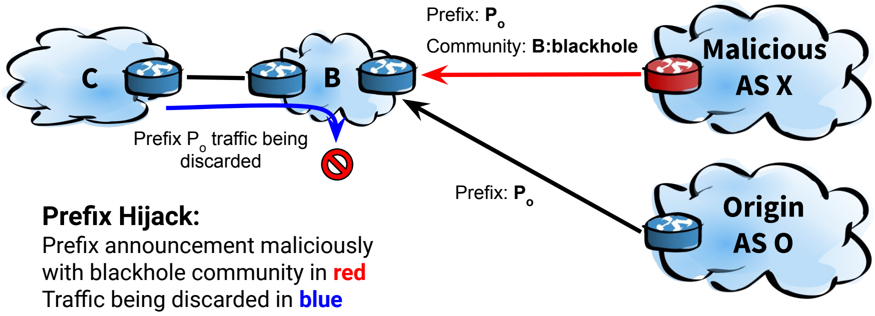

Prefix: P_o
Community: B:blackhole
Malicious AS X
C
B
Prefix P_o traffic being discarded
Prefix: P_o
Origin AS O
Prefix Hijack:
Prefix announcement maliciously with blackhole community in red
Traffic being discarded in blue

Figure 3.1: Origin AS O advertises its prefix. A malicious AS X injects a blackhole community for the same prefix, causing AS B to drop traffic destined for the prefix. This results in AS C being unable to reach the origin AS O.

Although the first improves route visibility, its effectiveness as an attack vector is limited because it does not directly trigger actions on remote networks. However, attackers can exploit the second method if they understand the specific meaning of the action communities. Our work does not provide the specific semantics of the action communities, so a malicious actor would have extra work to find them out.

3.6 Inference of AS Relationships

Over the past two decades, researchers have proposed several techniques to infer relationships between ASes [29, 37, 41, 50]. These techniques often rely on the assumption that BGP paths follow the *valley-free* property, meaning a path consists of zero or more customer-to-provider (c2p) links, followed by zero or one optional peer-to-peer (p2p) link, and then zero or more provider-to-customer (p2c) links [29]. However, a key challenge lies in the limited availability of data on these business relationships. Autonomous systems rarely disclose this information, hindering the ability to accurately annotate the Internet's AS graph. This lack of data complicates the deployment of many network applications, such as congestion detection between specific peering partners (*e.g.,* identifying congestion detection between ASes with specific peering agreements [19]), malicious AS identification, and even the deployment of BGP security mechanisms [33, 57, 87].

Jin *et al.* [50] propose a probabilistic algorithm (ProbLink) to detect complex relationships between ASes, which are characterized by non-obvious routing paths and unconventional peering practices. While existing algorithms, such as AS-Rank, excel at identifying standard relationships that follow expected patterns, they struggle with these complexities. ProbLink shares similarities with our approach, as it also uses the CAIDA AS-to-Org dataset [9] to identify sibling ASes. However, unlike our method, ProbLink does not use community information or consider the location of vantage points in its relationship inference, as we explain in Chapter 4.

Giotsas *et al.* [37, 50] argue that AS relationships are more complex than traditional models capture. They propose algorithms to identify non-conventional peering practices, such as hybrid relationships (where ASes have different relationships at different peering locations) and partial transit relationships (where a provider's transit service is limited to its customer cone). Their analysis of data from March 2014 revealed that 4.5% (or 4,026) of the 90,272 provider-customer relationships were complex, with 1,071 hybrid and 2,955 partial transit. They also observed that while some relationships are easy to infer, others are more challenging to determine.

Feng *et al.* [26] infer links between ASes that are difficult to detect because of the uncertainty of the relationships. They show that changes in the coverage of the route collectors when adding or removing new collectors can lead to uncertainties in the inference process. Some other efforts [75, 100, 104] propose or discuss the use of BGP communities to infer AS relationships and show that they enable better accuracy.

We present in Chapter 4 a heuristic for detecting the existence of sibling ASes on a set of route announcements from BGP route collectors. The algorithm uses CAIDA's AS-to-Org database [9, 74] to detect sibling ASes, but does not rely on AS relationship inferences. The approach, however, detects the relationship's existence without identifying the ASes involved. In Chapter 5 we also uses data from route collectors but goes beyond detection and identifies the ASes squatting the communities of other ASes, which can indicate sibling ASes or some other agreed-upon relationship between the ASes.

A recent work [74] shows that the state-of-the-art algorithms for inferring AS relationships lack ground truth validation and present similar results of the evaluation for regions like ARIN and LACNIC, but with validation that covers only 31% for the first and less than 1% for the second. Other research aims to infer sibling relationships using multiple data sources provided by network operators, such as IRR, websites, public documents, and PeeringDB [3, 12]. The difference from our algorithm for AS squatters in Chapter 5 is that we infer AS relationships in the real world using publicly available data.

3.7 Summary

This chapter examines the significant effort that the research community has invested in determining the semantics of BGP communities and understanding their use by network operators. BGP's flexibility, driven by politics and agreements rather than fixed rules, allows operators substantial freedom in modifying export and import routing policies. However, this flexibility also complicates the task of accurately interpreting the usage of BGP communities, especially when dealing with outdated information.

In the following chapters, we detail our approaches to inferring location and action communities. In Chapter 4, we present our contributions to inferring BGP communities related to locations, and in Chapter 5, we introduce a method for correlating ASes using informational communities to improve the identification of action communities. Our work is a step toward helping both the research community and network operators understand and define the usage of communities in real-world scenarios.

Chapter 4

Location Communities

It is always possible to agglutinate multiple separate problems into a single complex interdependent solution. In most cases this is a bad idea.

– RFC 1925, *The Twelve Networking Truths*

In this chapter, we present an algorithm for inferring location communities from publicly available BGP dumps collected by different projects, such as RouteViews, RIPE RIS, and Isolario. Recall that a location community is a tag that identifies the location (*e.g.,* city, country, continent, router, PoP, link, or interconnection) where a route was learned. We also present a set of heuristics to improve our algorithm's precision and filter out noises introduced by misbehaving ASes.

We evaluate the performance of our algorithm and the effectiveness of our heuristics using the CAIDA dataset of BGP communities and a manually-built ground-truth dataset from Tier-1 and Tier-2 ASes that publicize the semantics of their communities.

4.1 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 [2, 30, 37, 50, 64]. We first provide an overview of the key ideas in our inference algorithm using the example in Figure 4.1 (§4.1.1) and then present our algorithm formally (§4.1.2).

4.1.1 Overview

Consider a target AS T that tags received routes with location communities (see Figure 4.1). 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 4.1), 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 [39], 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 a route with AS path $\langle N_3, N_4, N_5 \rangle$ (blue dashed line in Figure 4.1). Let $I_{T,3}, I_{3,4}$, and $I_{4,5}$ be the

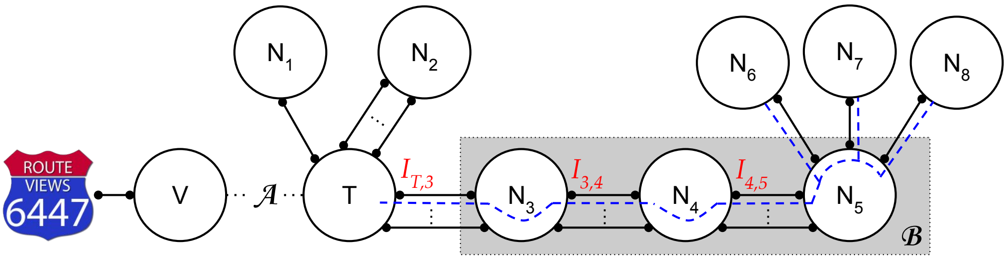


Figure 4.1: Example of how long sequences of ASes between origins and a target AS T constrains the set of locations of routes received and chosen by AS T. Example of how long sequences of ASes between origins and a target AS T constrain the set of locations of routes received and chosen by AS T. We denote 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.

interconnections traversed by the route. 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 [79] and choose routes from N_3 received at a particular interconnection. For example, if N_3 prefers to receive traffic 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 [79]) 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 [37]. 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

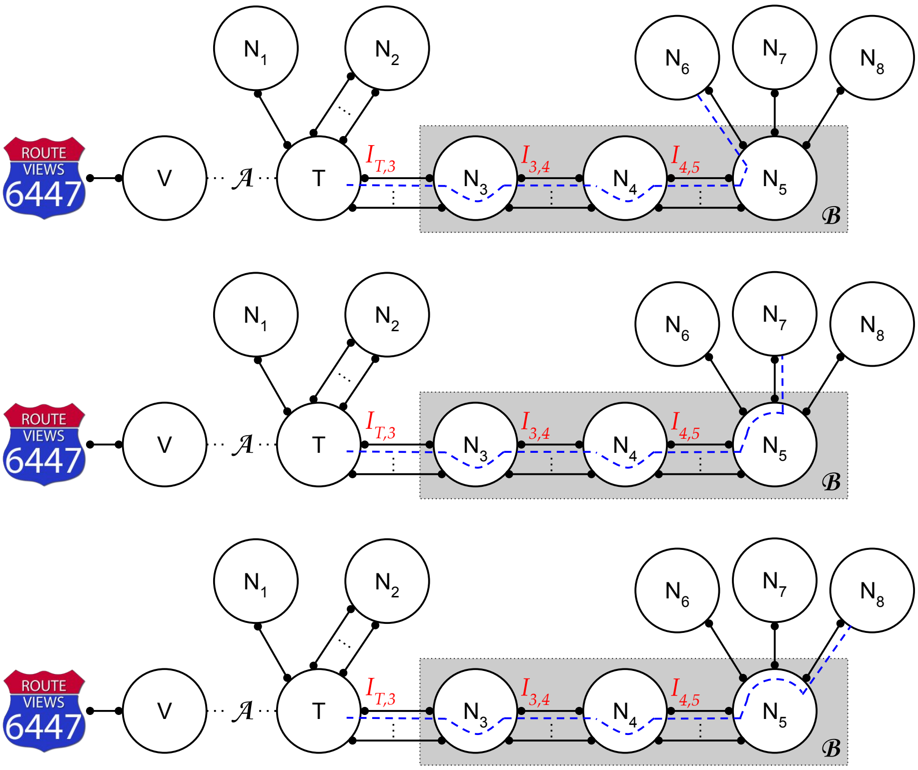


Figure 4.2: Example of how sequences of ASes with different origins and a target AS T constrain the set of locations of routes received and chosen by AS T.

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 other words, interconnection $I_{3,4}$ is *also* constrained by the interconnections between ASes N_3 and N_4 and their routing policies. The same constraints apply to $I_{4,5}$. The implication is that chosen routes traversing a 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 AS T to receive routes traversing $\langle N_3, N_4, N_5 \rangle$ 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

Table 4.1: 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

with traffic engineering communities by combining observations on multiple routes from different origins. Figure 4.2 illustrates the idea. In the figure example, the 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.

4.1.2 Inference Algorithm

Our algorithm looks for routes from multiple origins traversing an overlapping sequence of ASes before reaching a target AS T, 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 sequence of ASes between V and T, $\mathcal{B}$ is a sequence of ASes following T, and $\mathcal{S}$ is a suffix containing all ASes after $\mathcal{B}$ up to the origin AS. We consider that $\mathcal{B}$ constrains route propagation and the interconnections where AS T's chosen routes are received. $\mathcal{A}$ can be empty and AS V may be considered the target T, in which case $V = T$ and $\mathcal{A} = \emptyset$. For inferring communities, we require that $\mathcal{B}$ is nonempty, *i.e.,* contains at least one AS because we need to track the interconnection points between the ASes. 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 $|\mathcal{S}| = 1$.

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 4.1 summarizes the notation, and Algorithm 1 shows the pseudocode.

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 of AS T if $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ contains at least K_{origins} distinct routes. In other words, we require announcements by 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.)

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 [55] or ASes that remove BGP communities from announcements.[1] 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).

[1] BGP communities are a transitive attribute and ASes are not supposed to arbitrarily remove them from routes [11]. 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 [58, 59].

Algorithm 1: Inference of Location Communities

```
for each vantage point $v$ do
  $\mathcal{L}_v \leftarrow \emptyset$ {Set of location communities inferred from $v$'s routes}
  for each $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ in routes from $v$ do
    if $|\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})| < K_{\text{origins}}$ then
      continue
    end if
    $\mathcal{C} \leftarrow$ all communities from AS $T$ or of a sibling of $T$ appearing in
    $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$
    for each community $c \in \mathcal{C}$ do
      $N_c \leftarrow$ number of routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ with $c$
      if $N_c \div |\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})| \geq K_{\text{prev}}$ then
        $\mathcal{L}_v \leftarrow \mathcal{L}_v \cup \{c\}$
      end if
    end for
  end for
  for each community $c \in \mathcal{L}_v$ do
    $\mathcal{R}_c \leftarrow$ set of routes with $c$
    $\mathcal{R}_T \leftarrow$ set of routes whose AS paths traverse $c$'s AS or any of its siblings
    $\mathcal{F}_c \leftarrow \mathcal{R}_c \setminus \mathcal{R}_T$
    if size of the minimum hitting set of $\mathcal{F}_c \geq K_{\text{filter}}$ then
      $\mathcal{L}_v \leftarrow \mathcal{L}_v \setminus \{c\}$
    end if
  end for
end for
return $\bigcup \mathcal{L}_v$ for all vantage points $v$
```

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 4.2).

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).[2]

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.

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 $\mathcal{F}_c = \mathcal{R}_c \backslash \mathcal{R}_T$ 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 [31, 54], and we solve it using a greedy heuristic, which provides a tight approximation of the optimal solution [90].

[2] 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.

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 (§4.3), some communities are only visible from specific vantage points, so taking the union across collectors and vantage points maximizes coverage.

4.1.3 Implementation

Our implementation consists of over 2,100 lines of Python, with extensive use of the Pandas library for data processing. We use Snakemake [71] 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 [53].

4.2 Datasets

We use BGP feeds from RouteViews [68], RIPE RIS [81] and Isolario [42].[3] 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 MRT BGP updates could also be used, which would possibly increase the number of observed communities. Table 4.2 shows a summary for the route table dumps from December 2017 and 2020. We use CAIDA's

[3] We do not use PCH feeds [45] as they do not include BGP communities.

Table 4.2: 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

AS-to-Org database for identifying sibling ASes [9].[4] 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 4.3 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 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 [92]. 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 or 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 [7]. Our ground truth dataset includes 1.7 times more geolocation communities than CAIDA's database for the ASes in our dataset (not including autogenerated

[4] We built and evaluated an alternate sibling database by grouping ASes whose abuse contact e-mail have the same domain. We omit these results as they are quantitatively similar to those obtained with CAIDA's AS-to-Org database.

Table 4.3: Number of communities for ASes in our ground-truth dataset by type and geolocation communities in CAIDA's database [7].

	Community Type				CAIDA
Network (AS)	Geo	Dev/link	Relation	Action	[7]
Tier 1 [101]					
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 [102]					
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.

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.

4.3 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 (§4.3.1). We discuss community visibility in BGP dumps and how additional vantage points could improve recall (§4.3.2). We quantify the impact of each parameter on our algorithm's accuracy and show that inferences are not sensitive to parameter values (§4.3.3).

We compare our database of location communities with CAIDA's manually-built dataset and show we achieve competitive precision and significantly higher recall (§4.3.4). Finally, we present a characterization of the adoption and stability of location communities (§4.3.5).

4.3.1 Inference Accuracy

We quantify inference accuracy with precision and different views of recall [44]. *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 4.4 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 4.3.3. Table 4.4 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 4.4: Precision, recall, inferable recall (inf. recall), F1-score, and the number of inferred, correctly inferred, and inferred but undocumented (undoc.) 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.

			Inf.		Inferred Communities		
Configuration	Precision	Recall	Recall	F1 Score	Total	Correct	Undoc.
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

Table 4.5 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* [98] of 79% even if location and action communities were balanced.[5] We can increase the precision for action communities by tuning the algorithm's parameters (*e.g.,* prioritize precision).

4.3.2 Community Visibility and Recall

Figure 4.3 shows the cumulative distribution of the number of inferred location communities (left y-axis) and the number of inferred communities (right y-axis)

[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).

Table 4.5: Number of communities from ASes in our ground-truth dataset *seen* in BGP and *inferred* (inf) 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						Undocumented	
	Location		Relationship		Action			
Configuration	seen	inf	seen	inf	seen	inf	seen	inf
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

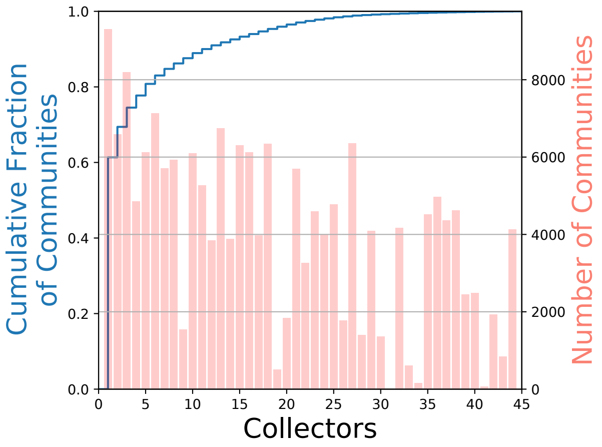


Figure 4.3: Number of inferred communities and recall as a function of the number of collectors.

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 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.

4.3.3 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.

Number of origins.

Figure 4.4 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

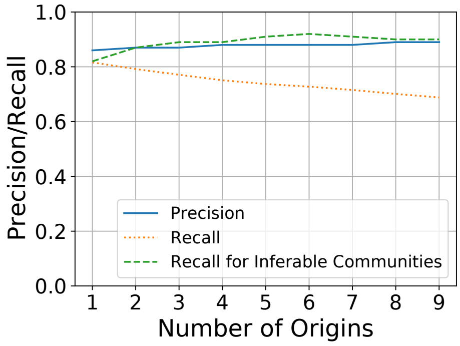


Figure 4.4: Precision and recall as a function of K_{origins}. High precision for $K_{\text{origins}} = 1$ indicates that origins rarely tag *all* their announcements with traffic engineering communities of other ASes.

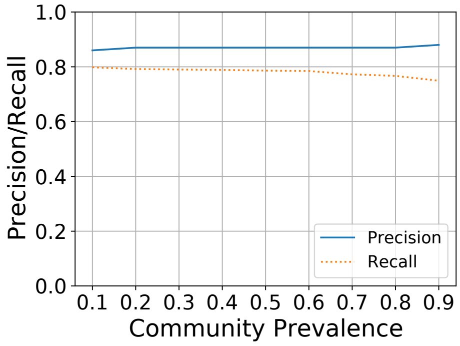


Figure 4.5: Precision and recall as a function of K_{prev}. Location communities appear on most routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$, so increasing K_{prev} up to 0.9 has small impact on precision and recall.

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 4.4 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.

Community prevalence.

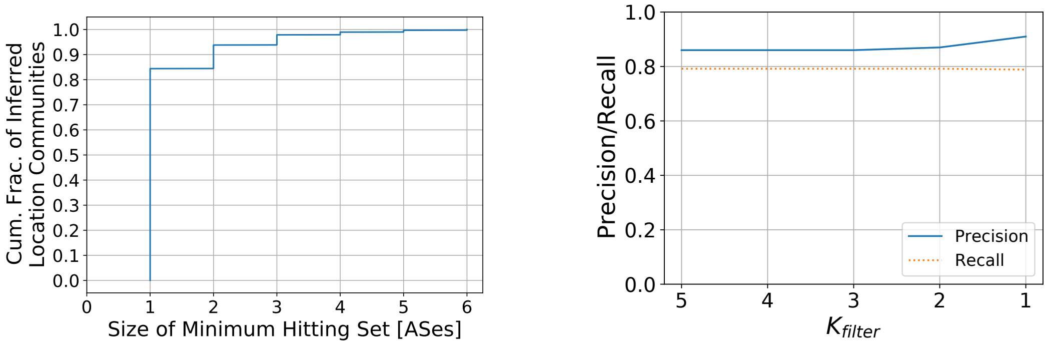
Figure 4.5 shows the impact of K_{prev}, the fraction of routes in $\mathcal{R}(V, \mathcal{A}, T, \mathcal{B})$ that a community needs to appear in to be inferred as a location community. Similar to Figure 4.4, we find that precision and recall are high and do not vary significantly as a function of 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.

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 4.6a 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 4.6b 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



(a) Distribution of minimum hitting set sizes

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

Figure 4.6: 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 4.6a, $x = 1$). 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 4.6b).

say an AS A uses a community from its sibling AS T when a community owned by T appears on a route that traverses A and does not traverse 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 indicates that siblings do share BGP communities, and accounting for this sharing is useful when filtering location communities.

Number of constraining ASes

Figure 4.7 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 4.1.1, 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 [16], 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

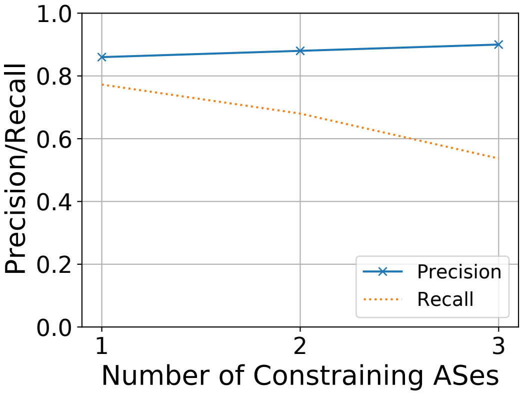


Figure 4.7: Impact of the number of constraining ASes, *i.e.*, $|\mathcal{B}|$, on recall and precision. More constraining ASes limit where chosen paths are received by a target AS T, improving precision, but fewer AS paths are long enough to support many constraining ASes, reducing recall.

can be configured to require more constraining ASes, which will lead to higher precision at the cost of recall.

4.3.4 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 4.6 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.

Table 4.6: 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

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.

4.3.5 Adoption and Stability of Location Communities

Figure 4.8 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 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 4.9 evaluates how stable are location community inferences over time. Figure 4.9(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 4.9(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

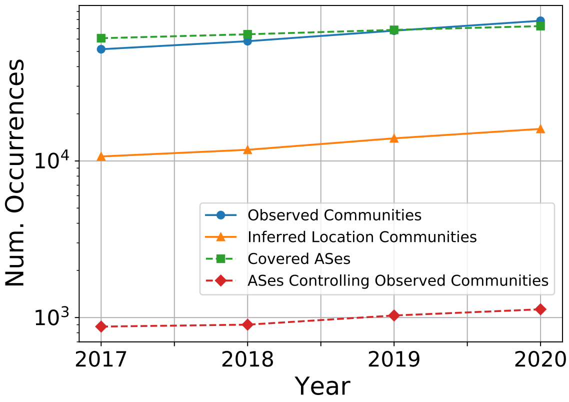
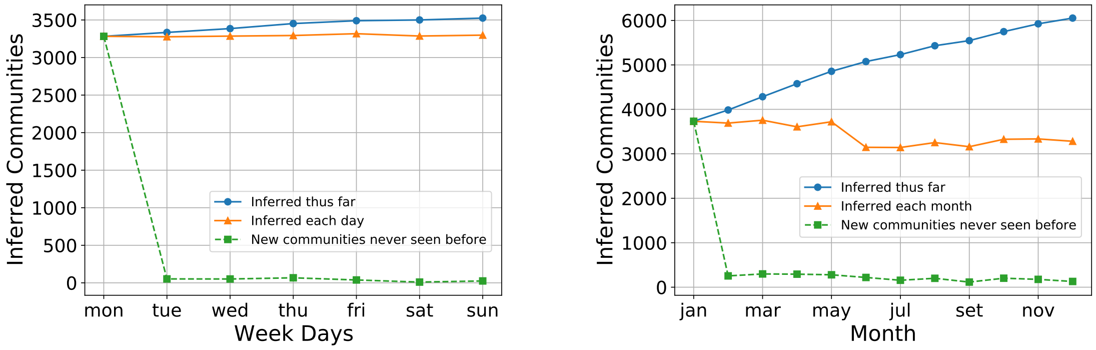


Figure 4.8: BGP community use in the Internet, quantified as the number of distinct BGP communities observed, number of inferred location communities, and the number of ASes controlling BGP communities.

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.



(a) First week of December 2020

(b) First day of each month in 2020

Figure 4.9: 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.

4.4 Summary

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 use 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. 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 [7], our inference algorithm generates a database with similar precision and much higher recall.

We identified 19.67% of the communities in 2020 as location communities. We make our database with 15,505 inferred location communities as well as our code publicly available [53].

Chapter 5

Action Communities

"One does not discover new lands without consenting to lose sight of the shore for a very long time."

– The Counterfeiters, *André Gide*

This chapter outlines the expected propagation patterns of action communities on the Internet, establishing a baseline understanding of how these communities should ideally propagate through the network. Then, we discuss the complexities and challenges of identifying action communities within an announcement that may contain hundreds of communities. We also examine common uses of action communities and instances in which their propagation deviates from the expected patterns.

Our research unexpectedly found evidence of ASes that consistently "squat" the information communities of other ASes. This suggests potential undisclosed agreements or relationships between these ASes. We evaluated the performance of our algorithm and the effectiveness of our heuristics using a manually built ground-truth dataset from ASes that make public the semantics of their communities.

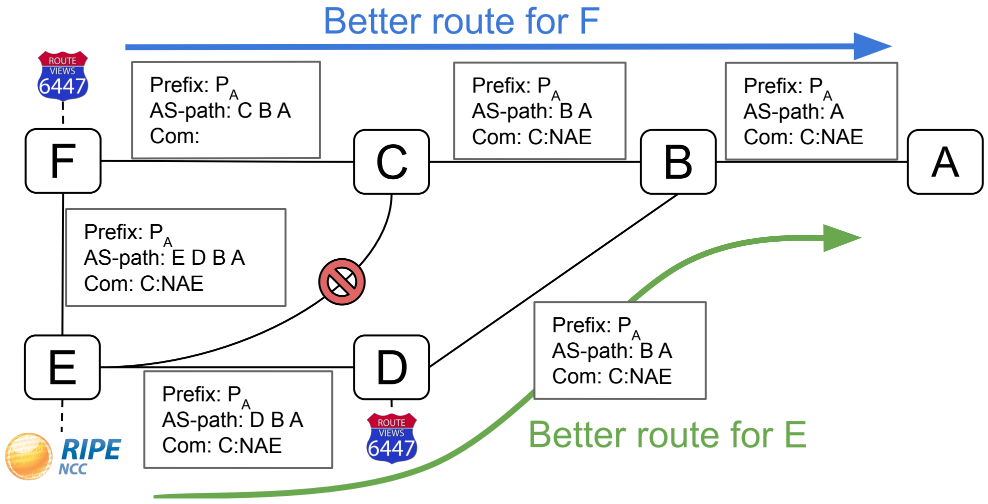


Figure 5.1: Example illustrating how an action community is more likely to appear in routes that do not include its controlling AS. The community C:NAE instructs AS C not to advertise routes to AS E. We can observe the community C:NAE on routes without AS C exported by ASes D, E, and F.

5.1 BGP Community Propagation

Given BGP community semantics, information communities should be tagged only on routes traversing their controlling ASes, as the controlling AS is the one that tags routes with the relevant information [89]. For example, an information community X:Y specifying that AS X received a route from a customer and a community X:Z specifying that AS X received a route in Europe can only be meaningfully added to a route by AS X.

On the other hand, action communities are less likely to be tagged on routes after traversing their controlling ASes due to multiple factors we discuss next. Figure 5.1 illustrates each factor; it shows propagation of a prefix P originated by AS A with action community C:NAE, which asks the controlling AS C to **n**ot **a**dvertise the route to AS **E**. Such a community could be used, for example, to steer traffic from AS E through AS D for load balancing or performance reasons.

1. An action community X:Y is added to a route by other ASes to request that AS X takes action Y. A route tagged with X:Y may be received by other ASes and exported to BGP collectors without traversing AS X. In Figure 5.1, AS A

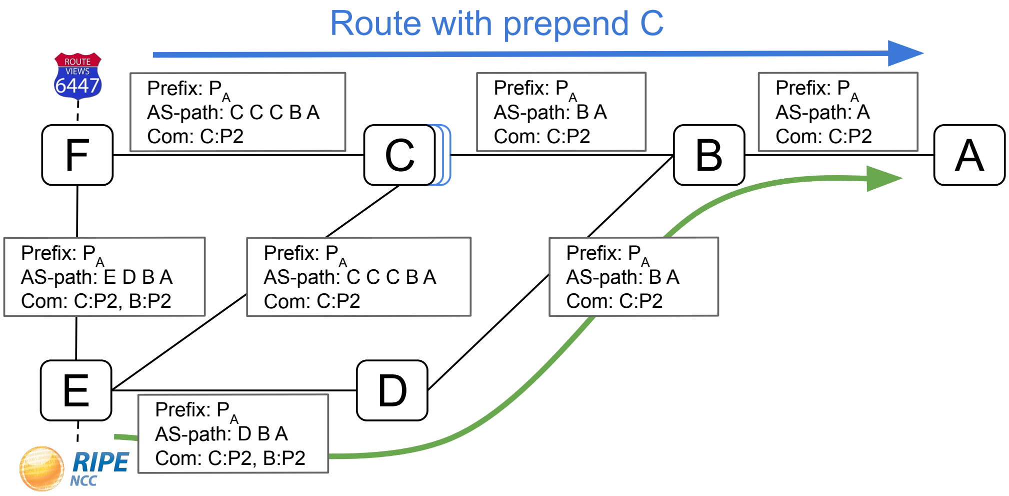


Figure 5.2: Example illustrating scenarios where action communities may appear in routes traversing their controlling ASes. AS C does not remove its action communities from routes after taking the requested action, and AS D adds an action community for AS B in routes that have already traversed AS B.

added the action community to its announcement. The route propagates, carrying the community, and is exported to a collector by AS D without traversing the controlling AS C.

2. Many action communities make routes less preferable by making them longer (prepending), reducing their preference (set LocalPref), or directly restricting propagation (no-advertise). As a result, routes with action communities that traverse the controlling AS are less likely to propagate compared to routes that avoid the target AS. In Figure 5.1, AS E does *not* receive a route from AS C, leading AS E to choose the route received from AS D, which does not traverse AS C.

3. An action community has no use for ASes other than the controlling AS after the requested action has been taken, so ASes often remove their action communities from routes before propagating them [59, 93]. In Figure 5.1, AS C removes the community from the route it announced to AS F, which chooses a route through AS C that does not carry the action community.

5.2 Challenges

Although we expect action communities not to be tagged on routes traversing their controlling ASes, this is not always true. Several factors may lead to action communities being tagged on routes traversing their controlling ASes, making their identification challenging. Figure 5.2 illustrates some scenarios on routes for a prefix P announced by AS A.

1. The controlling AS may take action on an action community and not untag it from the route due to unintended BGP configuration or by design (when the operator willfully propagates action communities). In Figure 5.2, AS C does not untag action communities from routes after taking the requested action. AS B tags community C:P2 asking AS C to prepend itself twice to the AS-path, and the community is observed with the controlling AS C on the route exported by AS F to the collector. If these ASes propagate their action and information communities equally, then our inference algorithm may be penalized in accuracy and recall.

2. The issue above is aggravated when the controlling AS does not act upon receiving an action community because of router misconfiguration or depending on the relationship with the neighboring AS from where it received the route, *e.g.,* an AS's routers may ignore action communities received from providers. In this case, the action community remains tagged on the route but does *not* reduce the route's preference; as route propagation is unconstrained, the route propagates broadly and causes the action community to be widely observed on routes traversing its controlling AS.

3. An AS may uselessly tag a route with an action community *after* the route has traversed the controlling AS, which has no impact on the route itself but may happen depending on how the router is configured. In Figure 5.2, AS D adds community B:P2 uselessly asking AS B, which is already in the path and will not receive the community, to prepend AS B twice to the AS-path. The community

B:P2 is observed with the controlling AS B on the route exported by AS E to the collector.

4. An operator may define *non-standard* BGP communities, where the first 2 bytes are set to a value different than the controlling AS's number. For example, AS9002 (RETN) uses community X:65533 as an action community that asks "prepend AS9002 three times when exporting the route to AS X." In this case, our algorithm would correctly infer the action communities but associate them with incorrect controlling ASes.

5.3 Identification of BGP Action Communities

In this section, we describe practical uses of BGP communities that violate the three factors described in Section 5.1 and complicate the inference of action communities (§5.3.1). We then describe how we identify communities that rarely appear with their controlling ASes as action communities (§5.3.2) and how we use them to uncover other action communities that do not necessarily satisfy our premise of appearing in route announcements without their controlling ASes (§5.3.3).

5.3.1 Identifying BGP Community Squatting

We observe that ASes may use BGP information communities defined by or belonging to other ASes. As an AS X is not supposed to tag routes with AS Y's information communities, we refer to this type of use as *squatting.* A common case is ASes using communities defined by one of their *siblings*, *i.e.,* another ASN under the control of the same organization [12, 29]. This behavior seems particularly common after network mergers and could result from the homogenization of routing policies defined using BGP communities across the merged ASes. For example, we observe routes traversing AS3549 (Global Crossing, acquired by Level3/Lumen [69]) tagged with several communities from AS3356 (Level3/Lumen); routes traversing AS286 (KPN, acquired by GTT [6]) tagged with communities from AS3257 (GTT); routes traversing AS5607 (British Sky Broadcasting, BSB) tagged with communities from AS4589 (Easynet, owned by BSB between 2006–2010 [103]).

As a result, a BGP AS-path traversing a set of ASes $\mathcal{S}$ may include communities belonging to other squatted ASes. This leads to information communities appearing in routes that do not traverse the controlling AS, which violates our intuition that only action communities will appear in routes without their controlling AS.

Inference Algorithm.

We propose an algorithm to infer ASes that squat another AS's communities. Our goal is to identify an AS X that systematically tags routes with BGP *information* communities whose first 16-bits is another AS Y. The challenge lies in differentiating between *(i)* an AS X squatting AS Y's information communities from *(ii)* an AS X simply using AS Y's action communities. We address this challenge by assuming that action communities are used selectively for specific, generally short-term, traffic engineering policies. In contrast, information communities are consistently applied after being defined, as routes are automatically tagged when an announcement traverses a router. Thus, we identify an AS X that consistently appears with AS Y's communities as a potential squatter.

We identify squatting AS-pairs using the routes from each RIPE RIS and Route-Views collector separately and then aggregate the inferences. Alternate approaches may be possible given different inference mechanisms; our approach strikes a compromise between obtaining enough routes for inferences, combining routes from all ASes peering with each collector, while trying to capture route properties specific to the view of the Internet's topology captured by that collector [72, 74].

For instance, one collector might be unable to identify that AS X squats the communities of another AS Y because an intermediate AS Z strips the squatted communities tagged by AS X. Another collector may observe routes with AS X's squatted communities if its routes do not traverse AS Z.

Our algorithm uses only publicly available information from RouteViews and RIPE RIS collectors. Consider the following notation:

- $\mathcal{C}(y)$ is the set of routes tagged with at least one community from AS Y;
- $\mathcal{R}(x)$ is the set of routes that traverse AS X; and
- $\mathcal{R}(\neg y)$ is the set of routes that do not traverse AS Y.

We check if an AS X is related to another AS Y by computing the following three metrics for each pair of ASes:

Coverage. Among the routes that do not traverse AS Y but are tagged with a community from AS Y, we compute the fraction that traverse AS X. More precisely, we define *coverage* $C(x, y) = |\mathcal{R}(x) \cap \mathcal{R}(\neg y) \cap \mathcal{C}(y)| \div |\mathcal{R}(\neg y) \cap \mathcal{C}(y)|$. Coverage is high when AS X appears in most of the routes tagged with AS Y communities even though they do not traverse AS Y. This implies that AS X "explains" most of the unexpected observations of AS Y's communities and could be squatting. Coverage is low when there are many routes unexpectedly tagged with AS Y's communities that cannot be attributed to AS X. Figure 5.3 illustrates the idea behind the coverage parameter. In the example, a subset of announcements shows that ASes 29140, 286, and 3356 appear with the community 3257:8794, which belongs to AS3257. At this stage of the algorithm, the three ASes in the yellow boxes are candidates for potential squatters of the communities from AS3257.

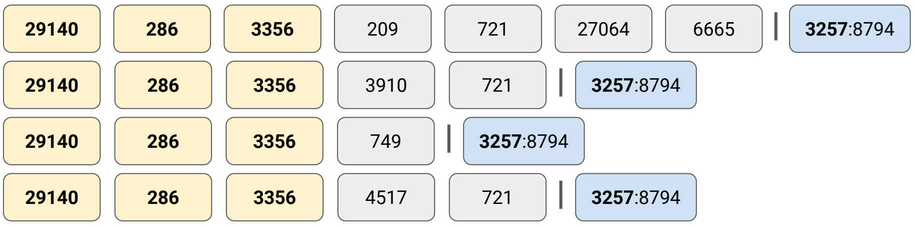
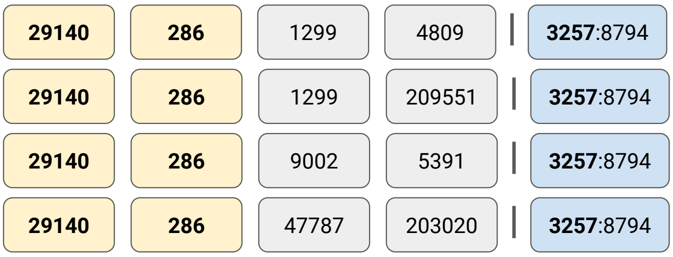


Figure 5.3: Subset of route announcements showing ASes coverage relative to the communities belonging to AS3257. All ASes in the yellow boxes are candidates for potential squatters.

Local Prevalence. Among the routes that traverse AS X but do not traverse AS Y, we compute the fraction tagged with a community from AS Y. More precisely, we define *local prevalence* $P_{\text{local}}(x, y) = |\mathcal{R}(x) \cap \mathcal{R}(\neg y) \cap \mathcal{C}(y)| \div |\mathcal{R}(x) \cap \mathcal{R}(\neg y)|$. Local prevalence is high when most routes traversing AS X are tagged with a community from AS Y even when the routes do not traverse AS Y. This implies AS X may be squatting and using AS Y's communities as its own information communities. Prevalence is low when many routes traversing AS X do not have a community from AS Y, which indicates AS X is not systematically squatting AS Y's communities: AS X may be simply using AS Y action communities or another AS on some routes traversing AS X is tagging them with AS Y's communities. Figure 5.4 presents a complementary subset of announcements to those in Figure 5.3. This subset shows that, based on local prevalence, AS3356 does not meet the squatting criteria: some announcements contain the AS3257 community without the AS3356 in the AS path. This indicates that AS3356 is not squatting AS3257's communities. At this stage of the algorithm, only ASes 29140 and 286 remain as potential squatters.



29140	**286**	1299	4809	|	**3257**:8794
29140	**286**	1299	209551	|	**3257**:8794
29140	**286**	9002	5391	|	**3257**:8794
29140	**286**	47787	203020	|	**3257**:8794

Figure 5.4: Subset of announcements showing the local prevalence of ASes in relation to the community of AS3257. In this case, AS3356 is no longer a candidate in relation to Figure 5.3, leaving only ASes 29140 and 286 as candidates for potential squatters.

Global Prevalence. Among the routes that traverse AS X, we compute the fraction that do not traverse AS Y but are tagged with a community from AS Y. More precisely, we define *global prevalence* $P_{\text{global}}(x, y) = |\mathcal{R}(x) \cap \mathcal{R}(\neg y) \cap \mathcal{C}(y)| \div |\mathcal{R}(x)| \leq P_{\text{local}}(x, y)$. Global prevalence is low when the supporting evidence that an AS is squatting is small compared to the number of routes observed through that AS. For example, AS X may appear on many routes through AS Y, which may not remove action communities from routes it propagates to AS X. Alternatively, AS X may be close to a BGP collector and appear on most collected routes, which may contain AS Y's action communities tagged by other ASes. Figure 5.5 illustrates the final stage of the inference algorithm. Using global prevalence, the algorithm excludes ASes primarily associated with exporting announcements to collectors, as they do not actually squat the communities of a given AS. Through the filtering steps shown in Figures 5.3 and 5.4, ASes 3356 and 29140 were removed by the local and global prevalence filters, respectively, leaving only AS286. Consequently, the inference results indicate that AS286 squats the communities of AS3257. In this specific example, AS286 (KPN) and AS3257 (GTT) are sibling ASes, as KPN was recently acquired by GTT [6].

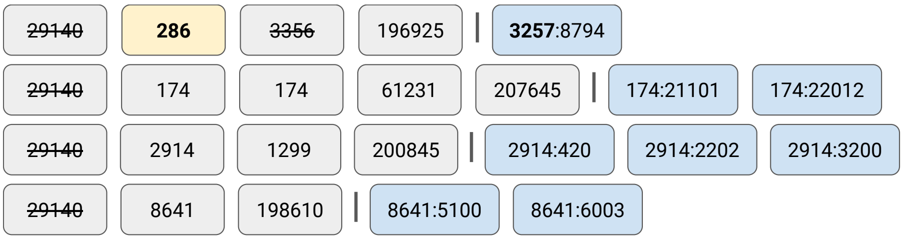


Figure 5.5: This subset of announcements illustrates the application of global prevalence, with hatched markings indicating ASes removed during the inference process. AS29140 appears in numerous announcements with various communities, which leads to its exclusion based on the global prevalence parameter. The algorithm's final inference identifies only AS286 as a potential squatter of the AS3257 communityies.

To infer if an AS Y is squatted by other ASes, we check if another AS X has coverage $C(x, y) > 0.9$, local prevalence $P_{\text{local}}(x, y) > 0.7$, and global prevalence $P_{\text{global}}(x, y) > 0.3$ (§5.5.1). To avoid inferences with weak support and possibly caused by noise in the BGP dumps, we also require that AS X appears squatting at least two communities from AS Y and that these communities are observed in at least six routes each. We justify these choices in Section 5.5.1. If multiple ASes are identified as possibly squatting AS Y's communities, we select the one with the largest coverage, largest local prevalence, largest global prevalence, or appearing furthest away from the route collector, in order. The high required coverage of 0.9 allows for at most one squatting relationship with a target AS Y from each BGP collector, but multiple squatting relationships with the same AS Y can be identified across multiple collectors.

When handling squatting relationships, we consider that the inferred relationships are bidirectional and transitive, such that if ASes A and B squat communities from AS C, we consider that ASes A, B, and C are part of one squatting relationship.

Special Cases

Manual inspection of the identified squatting AS-pairs indicates that some pairs are likely caused by typing errors. For example, we observed a community 15**9**85:9999 on paths traversing AS15**8**95, which leads to inferring AS15**9**85 as squatting AS15**8**95. We ignore a squatting relationship between two ASes when their ASNs have five digits and the ASNs have an edit distance of 1. We consider edit operations of substituting one digit for another or reordering two consecutive digits. We ignore all communities involved in these squatting relationships when inferring action communities to avoid errors. We consider only five-digit communities because typos are more likely to occur in longer character sequences [86] and are more challenging for an operator to detect visually. This choice is conservative, as typos in shorter communities may decrease the precision of our algorithm. However, this length is not a fundamental limitation of the approach and can be adjusted if necessary.

We also found some squatting AS-pairs likely caused by an integer overflow when 32-bit ASNs are used with classic 32-bit communities that store ASNs in just 16 bits. For example, we identified many communities from AS303 on routes traversing AS327983, where 303 = `0xffff & 327983`. We ignore all squatting relationships where the squatter ASN's last 16 bits are identical to the squatted ASN, and ignore all such communities when inferring action communities.

We also ignore all squatting AS-pairs involving an IXP ASN, as identified in

CAIDA's AS-relationship database [8]. Many IXPs define action communities to control announcement propagation through route servers (*e.g.,* [1]), but IXP route servers do not add their ASN to propagated routes, which may lead to some ASes being identified as squatting the IXP's communities.

5.3.2 Inferring BGP Action Communities

Our inference algorithm centers around checking how often a community is tagged on a route that does not traverse the controlling AS or any of its squatters, from now on collectively referred to as *controlling ASes*. Earlier in this chapter, we discussed the main difficulties in identifying action communities, enforcing a requirement that a community never appears with its controlling ASes is too restrictive. We design and evaluate different approaches to account for lack of visibility and noise in observed community usage. Algorithm 2 presents pseudocode covering all approaches.

Handling squatting ASes. We use the sets of squatting ASes identified in §5.3.1 to avoid inferring communities squatted upon as action communities. We compute the squatters for the same collectors used to infer action communities resulting in different AS relations. These relations will be used during the inference of the action communities. Before we execute our algorithm, we rewrite ASNs with squatting relationships when they appear in a route's AS-path or communities. In particular, we rewrite each ASNs with the smallest ASN among its set of squatting ASes (Line 2). This ensures that if a route traverses a squatting AS X and is tagged with a community from a squatted AS Y, then both ASNs will be rewritten with the smallest ASN in their set of squatting ASes. This effectively prevents identifying squatted communities as action communities.

Filtering Low-Visibility Communities We do not make inferences for communities that have limited visibility in public BGP dumps. We require that a community c is observed by at least two collector peers, and that each collector peer observes the community in at least four routes (counted in N^c_{vps}, Line 7, and verified in $\mathcal{C}_{\text{candidates}}$, Line 24). These thresholds are chosen empirically (§5.5.1); however, we show that inferences are not sensitive to their values as long as they are large enough to remove the long tail of rarely-seen communities from the inference process. This filter removed 11,836 communities from our inferences, representing less than 11% of the communities on BGP dumps. Our algorithm would be able to classify these communities if their use and visibility became more widespread.

Inferring Action Communities Our algorithm operates on each community independently (Line 5). For each community, our inference relies on computing the fraction of routes tagged with a community from AS Y that do not traverse AS Y. This is done by counting the number of routes with each community c (Line 6) and the number of these routes that do not traverse any of c's controlling ASes (Lines 8–10). Using these variables,

we infer as action communities those that are mostly *absent* from routes traversing their controlling ASes ($\mathcal{C}_{\text{absent}}$, Line 25). This approach allows some occurrences of the controlling ASes and accommodates errors and unexpected cases, like when an action community is not acted upon, *e.g.,* because it was not set by a customer of the controlling ASes, and remains tagged on the route after traversing the controlling AS.

Handling prepend communities Action communities that ask an AS Y to prepend itself to the AS-path *will* appear on routes traversing AS Y (prepended multiple times) if AS Y does not remove action communities from announcements. To allow the detection of prepend communities in these scenarios, we count the number of times a community appears on routes with AS-paths that have the community's controlling ASes prepended (Lines 13 and 26). This approach has the negative side-effect of possibly inferring some information communities that often appear on routes prepended with the respective controlling ASes as action communities.

Handling action communities added after the controlling AS An action community has no use after the controlling AS has taken the requested action. However, an AS may (uselessly) tag a route with an action community *after* it has traversed the controlling AS, which has no impact on the route itself but may occur depending on when the tagging is performed. These behaviors directly impact our inferences, as they make action communities more likely to appear on routes traversing controlling ASes and, thus, harder to differentiate from information communities. To filter this case, we use only uphill AS-paths, *i.e.,* AS-paths composed entirely of customer-to-provider relationships starting from the origin AS ($\mathcal{C}_{\text{before}}$, Lines 16–21 and 27). Our intuition is that customers often use action communities to control how providers handle their announcements; thus, a community c tagged on an *uphill* AS-path traversing c's controlling ASes is less likely to have been tagged after the controlling AS and more likely to be an information community.

Handling ASes that do not remove action communities from route announcements An action community has no use after the controlling AS has taken the requested action. However, the controlling AS is not required to untag the action community from the route. To sidestep the uncertainty added by ASes that do not remove action communities, we apply a relaxation filter allowing the community to appear with its controlling AS in a small fraction F of the announcements in each selected vantage point (Lines 25–27).

5.3.3 Uncovering Missing Action Communities

Our inference algorithm requires a minimum number of announcements carrying a community to classify it as an action community with high confidence. However, route collectors do not provide complete coverage of the Internet routes, and some ASes filter all communities before forwarding route announcements, impacting the communities' visibility and our algorithm's recall. To circumvent this limitation, we

Algorithm 2: Inference of Action Communities

```
Input: $\mathcal{R} \leftarrow$ set of all routes, each with AS-path $P$ and set of communities $\mathcal{C}$.
Requirement: AS-paths and communities rewritten with each ASN mapped to the lowest ASN in its set
  of squatting ASes, if any.

for each route with rewritten AS-path $P$ and set of communities $\mathcal{C}$ in $\mathcal{R}$ do
    $\mathcal{C}_{\text{global}} \leftarrow \mathcal{C}_{\text{global}} \cup \mathcal{C}$    {Track all communities visible in BGP dumps.}
    for each community $c$ in $\mathcal{C}$ do
        $N^c_{\text{routes}} \leftarrow N^c_{\text{routes}} + 1$    {Count routes tagged with community c.}
        $N^c_{\text{vps}}[P_0] \leftarrow N^c_{\text{vps}}[P_0] + 1$    {Count routes exported by BGP collector peer $P_0$ tagged with community c.}
        if $c$'s controlling ASes $\notin P$ then
            $N^c_{\text{absent}} \leftarrow N^c_{\text{absent}} + 1$    {Count routes tagged with community c that do not traverse c's controlling ASes.}
        else if $P$ is uphill then
            $N^c_{\text{info}} \rightarrow N^c_{\text{info}} + 1$    {Count routes tagged with community c that traverse c's controlling ASes on uphill path.}
        end if
        if any of $c$'s controlling ASes is prepended in $P$ then
            $N^c_{\text{prepended}} \leftarrow N^c_{\text{prepended}} + 1$    {Count routes tagged with community c with its controlling ASes prepended.}
        end if
        if $P$ is uphill then
            $N^c_{\text{uphill}} \leftarrow N^c_{\text{uphill}} + 1$    {Count uphill routes tagged with community c.}
            if $c$'s controlling ASes not in the customer cone of ASes in $P$ then
                $N^c_{\text{before}} \leftarrow N^c_{\text{before}} + 1$    {Count uphill routes terminating before c's controlling ASes.}
            end if
        end if
    end for
end for

$\mathcal{C}_{\text{candidates}} \leftarrow \{c \mid c \in \mathcal{C}_{\text{global}} \land |N^c_{\text{vps}}| \geq 3 \land \min(\text{values}(N^c_{\text{vps}})) \geq 4\}$
$\mathcal{C}_{\text{absent}} \leftarrow \{c \mid c \in \mathcal{C}_{\text{candidates}} \land (N^c_{\text{absent}}/N^c_{\text{routes}}) \geq 1 - F\}$
$\mathcal{C}_{\text{prepend}} \leftarrow \{c \mid c \in \mathcal{C}_{\text{candidates}} \land (N^c_{\text{absent}} + N^c_{\text{prepend}})/N^c_{\text{routes}} \geq 1 - F\}$
$\mathcal{C}_{\text{before}} \leftarrow \{c \mid c \in \mathcal{C}_{\text{candidates}} \land (N^c_{\text{before}}/N^c_{\text{uphill}}) \geq 1 - F\}$
$\mathcal{C}_{\text{prefix\_tree}} \leftarrow \text{PrefixTree}(C_{\text{absent}}, C_{\text{candidates}})$    {All communities that match the prefix-tree leaves tagged as action.}

Output: $\mathcal{C}_{\text{action\_communities}} \leftarrow \mathcal{C}_{\text{prepend}} \cup \mathcal{C}_{\text{prefix\_tree}}$
```

use the communities we infer with high confidence in Algorithm 2 to build a prefix tree from the decimal digits of the community labels and classify other communities with low visibility or that fall under the special cases we list in Section 5.3.2.

The rationale behind using a prefix tree is that a natural way for an AS to define its communities is by numbering communities of the same type sequentially and leaving some space between types to accommodate future expansions of the existing types. By following this pattern, communities of the same type share a common prefix, whose length can vary depending on the number of communities of the same type defined sequentially and the space between the types. We observe that most ASes on the Internet follow this pattern. Some use large blocks of fixed size for each type, while others use smaller blocks of variable sizes. Figure 5.6 shows a prefix tree for the communities documented by AS3257. A leaf, annotated with

A for action and I for information, indicates the type of communities that share the prefix starting at the root up to the leaf. For example, labels 3257:02XXX and 3257:1XXXX represent action communities, while 3257:08XXX and 3257:3XXXX represent information communities.

Specifically, we build a prefix tree for each AS that Algorithm 2 infers at least one action community. We treat the label of a community as a string with five digits (*i.e.,* the maximum number of decimal digits a 16-bit label can represent), filling in the string with zeros on the left when the label has fewer than five digits. Then, we divide the communities into sets containing communities with the longest common prefixes. We build one branch of the prefix tree for each set using only the digits in the longest common prefix of the communities in the set. As Algorithm 2 infers only action communities, all the leaves of the prefix trees are labeled with A. We apply the AS's prefix tree to all its communities that appear in $\mathcal{R}$, *i.e.,* the set of all routes from the BGP collectors we process, and classify the communities that share a prefix with a leaf as action communities.

We validated this idea using the communities of 15 ASes in our ground-truth dataset that have at least 20 communities. Specifically, we conducted experiments by building a prefix tree with a random subset of communities from an AS's ground truth and testing with the remaining communities from the same AS. We varied the subset sizes from 20% to 90% of the total communities and ran 100 experiments for each subset size. We measured the average precision and recall, with the average precision exceeding 99.5% for all subset sizes and the average recall ranging from 90.9% to 96.82%. These results indicate that the prefix trees effectively capture the structure of the community definitions of the selected ASes.

5.3.4 Implementation

Our implementation consists of over 2,354 lines of Python, with use of the NetworkX library for graph processing, regex for AS path evaluation, and `pickle` for serialization. We use Snakemake [71] to automate our database construction.

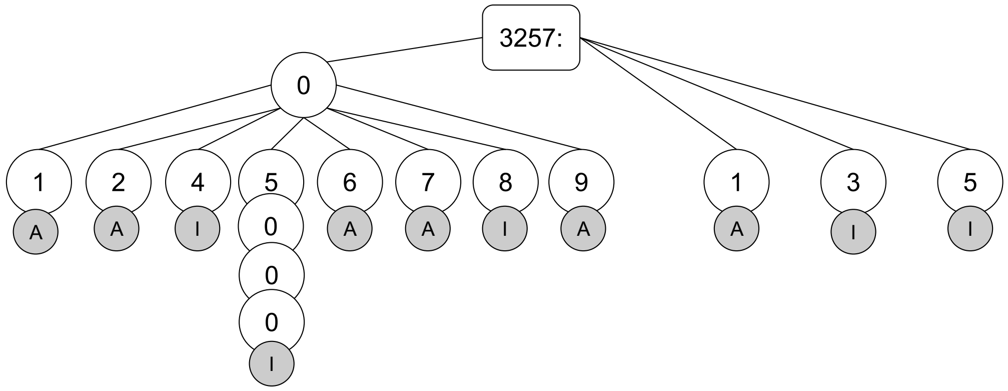


Figure 5.6: A prefix tree for the documented BGP communities from AS 3257. The branch 05000 is unusually long because it contains only one community, with no other communities sharing the 05* prefix.

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 [52].

5.4 Datasets

We evaluate our algorithms using the first BGP routing table (RIB) dumps of Dec. 1st, 2023, from all 55 BGP route collectors operated by RIPE RIS [81] and RouteViews [68]. We use bgpscanner [49] to process the RIB dumps and remove routes with AS-level loops (0.005% of routes) or AS-sets [61] (0.03%). For each route, we extract the prefix, the AS path, and the possibly-empty set of attached BGP communities.

We use CAIDA's AS-relationship database [37,64] from Dec. 1st, 2023, to identify the *uphill*, *peak*, and *downhill* regions of the AS-path. We ignore 0.27% of routes that violate valley-free routing and attempt to infer relationships for AS-pairs in a route missing from CAIDA's database. If the existing relationships are compatible with valley-free routing and at most one relationship is missing at the peak, we infer missing relationships as customer-to-provider in the uphill region, provider-to-customer in the downhill region, and peer-to-peer if there is a missing relationship at the peak.[1] We perform this inference of missing relationships for each route separately; inferences from one path do not carry over to other routes.

We parse public information from Internet Routing Registry (IRR), NL NOG [73], and OneStep [91] databases to extract ground-truth information to classify BGP communities according to their semantics. We use this ground truth dataset to evaluate the precision and recall of our inference algorithm. Our database includes information about the type of AS (*i.e.,* Tier-1, Tier-2, and others) of each community to evaluate how the performance metrics vary as a function of where the AS is on the Internet hierarchy. It contains 16,421 action communities from 74 ASes: 14322, 532, and 1567 from Tier-1, Tier-2, and other ASes, respectively. Although our ground-truth dataset contains a little over 1% of 6,158 ASes appearing on BGP communities in public BGP dumps, the ASes we consider are large and make more significant use of BGP communities than the average AS on the Internet. Overall, the ASes in our ground-truth dataset account for 16.8% of visible BGP communities in Dec. 2023. Also, our ground-truth dataset covers a variety of action communities, including selective advertisements, blackholing, prepending, and changing the LocalPref; with several ASes defining action communities that apply to specific peers or geographical locations.

To build the ground-truth dataset of ASes that squat the communities of other ASes, we also use public information about organizations, their ASNs, and their prefixes from the IRR databases. We use these databases to map ASNs to their controlling organizations and determine if two ASes are related by manually looking for similarities in organization names, geographical addresses, descriptions, and domain names for peering, operations, and abuse e-mail addresses [2,3,89]. To add relationships to the ground-truth dataset, we initially generated a set with the rela-

[1] This approach is equivalent to reapplying steps 5 and 11 of the original algorithm [64], but visiting ASes in the route from the peak toward the origin and from the peak toward the collector instead of following the transit and node degree gradients.

tionships that our algorithm for identifying squatting inferred with very restrictive parameters—*i.e.,* coverage = 0.9, local prevalence = 0.9, and global prevalence = 0.9—and manually classified the inferred relationships. We then gradually reduced coverage and local and global prevalence from 0.9 to 0.1 to increase the number of classified relationships until we could not validate the new ones.

We classified 59 relationships as confirmed and 23 as unconfirmed. To confirm a relationship, we used the similarities described previously. We consider a relationship unconfirmed if we find the documentation about the two ASes and it does not have any similar information that leads us to believe they are related. Note that this approach is conservative, as the ASes may be related even though the documentation does not reflect their relationship either by lacking the information or by being outdated.

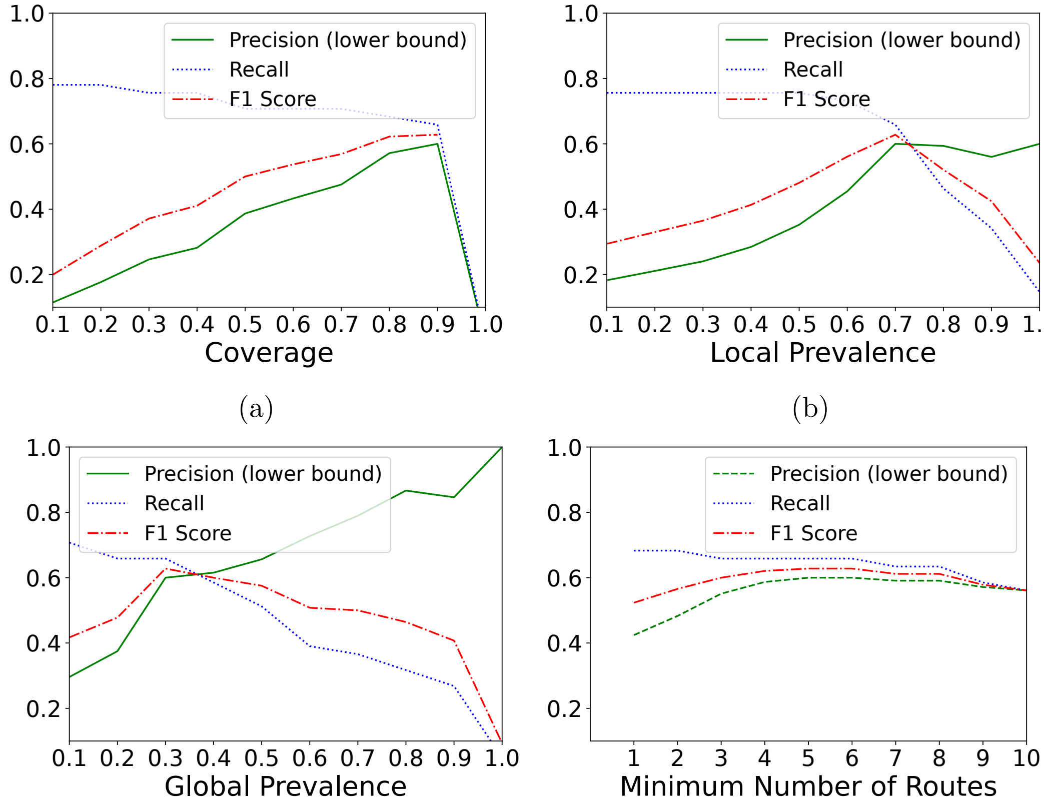
5.5 Evaluation

This section describes how we configure the parameters of our inference algorithms, evaluates the precision and accuracy of our inferences, and compares them with related prior work. We show that our algorithms are not strongly dependent on specific parameter configurations, *i.e.,* a broad range of configurations yields positive results. We make our datasets and evaluation code public to ease the replication of our results and independent executions of the inference algorithms [52].

5.5.1 Setting Parameters

Configuration of the Squatting Inference Algorithm

As described in Section 5.3.1, our algorithm for identifying squatters relies on three parameters: *coverage*, *local prevalence*, and *global prevalence*. These parameters are fractions in the interval [0, 1] computed over sets of routes. To determine the best parameters and investigate if they generalize to other datasets, we use route announcements from December 2022 to explore different combinations of the



(a)

(b)

(c)

(d)

Figure 5.7: Each of the graphs (a)-(c) shows the behavior of one of the parameters of our algorithm when we keep the other two at their default (best) values. Increasing threshold values improves precision at the cost of recall, as expected, and the default values represent the inflection points of the F1-score curves. Graph (d) shows the impact of the minimum number of routes communities must appear to determine a squatting relationship.

parameters. Specifically, we vary coverage, local prevalence, and global prevalence in the interval [0.1, 1] in steps of 0.1, resulting in 1000 (*i.e.*, 10^3) combinations. We validate the inferred squatting relationships computing the precision and recall for each parameter combination using the ground-truth dataset described in Section 5.4.

The combination of coverage = 0.9, local prevalence = 0.7, and global prevalence = 0.3 yields the highest F1 score, so we choose it as the default configuration[2].

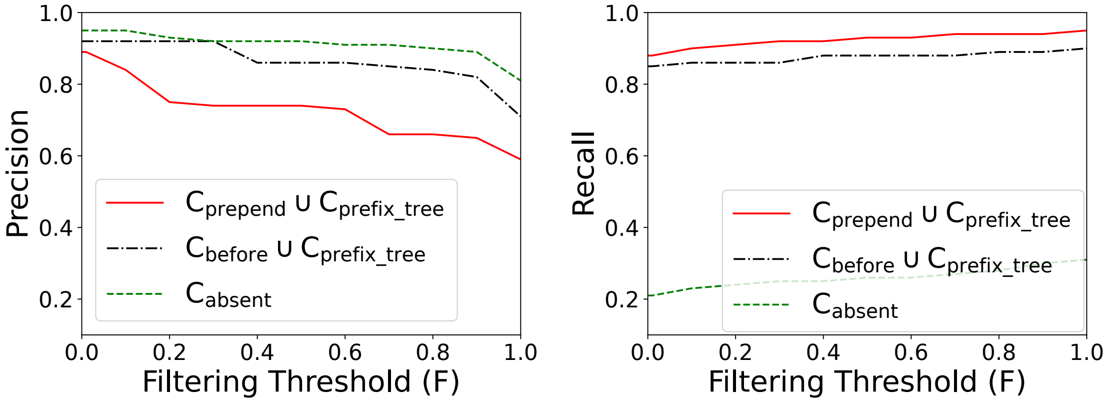
[2] We evaluated the Phi coefficient (also known as the Matthews Correlation Coefficient, MCC) [18, 66], and found that it is strictly higher than the F1-score, quantitatively similar to recall, and has no inflection point to aid in choosing default values for each parameter (not shown). While the Phi coefficient considers imbalance between classes, it is less suitable for our evaluation because the number of true negatives—AS pairs that have no squatting relationship—is exceed-

The best configuration achieves a lower bound on precision of 0.71 and a recall of 0.65. We note that 0.71 is a lower bound on precision because some of the inferred squatting relationships may be missing from our ground truth dataset (*i.e.,* we have not manually checked a pair of ASes); we take a conservative approach and report these inferences as incorrect, but some could be correct.

Figure 5.7 shows the precision, recall, and F1 score when we vary one parameter and keep the other two parameters fixed at their default (best) values. As expected, increasing threshold values improves precision at the cost of recall, and the selected values represent inflection points of the F1 score. We also observe that every parameter impacts the inferred relationships. Our algorithm infers no squatters when coverage = 1; thus, both precision and recall are zero.

Our algorithm inferred 54 pairs of squatting relationships, with 7 ASes appearing in multiple pairs, which we join for a final count of 48 (transitive) relationships. Of these relationships, the validated inferences include 26 sibling ASes, 2 neighboring ASes, 19 missing from our ground truth, and 7 unconfirmed.

We believe our automated inference of ASes squatting BGP communities might have applications for other studies relying on BGP communities (*e.g.,* validation of AS-relationship inference [37, 50, 64] and route change tracking [21, 36]). It might also benefit other efforts that seek to identify relationships between ASes. For example, the intersection of our community-based inference of squatters and our ground truth dataset contains five sibling relationships not identified by Chen *et al.*'s recent technique [12]. Finally, it is unclear why apparently unrelated ASes squat another's communities in some cases. We note that this practice, even if well-intended, may confuse troubleshooting efforts and policy filters not only for the ASes involved but also their neighbors [93].



(a) Precision (b) Recall

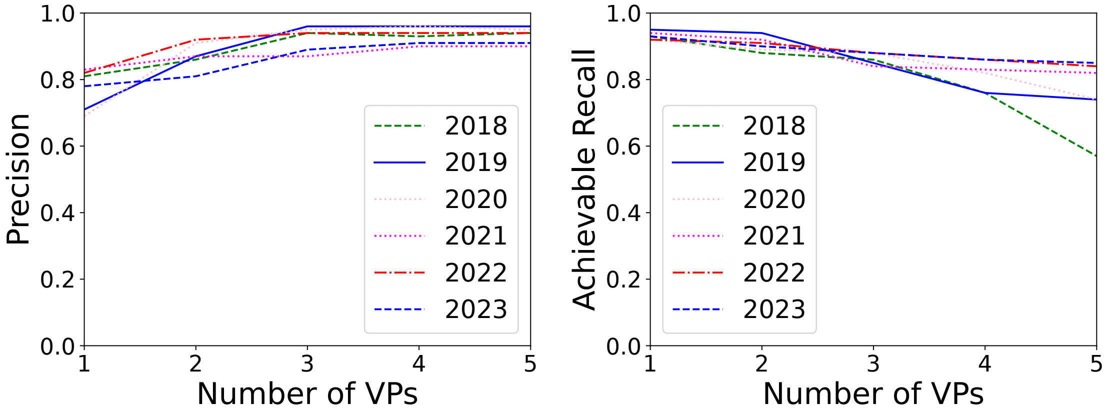
Figure 5.8: Inference performance as a function of the noise filter threshold F. Higher F values allow a BGP community to appear on more routes with its controlling AS and still be inferred as an action community.

Configuration of the Action Communities Inference Algorithm

We evaluate precision and recall for different community filtering thresholds (F in Algorithm 2). Figure 5.8a shows precision as we vary the filtering threshold on the x-axis, while Figure 5.8b shows the recall for the same configurations. We compare the more conservative $\mathcal{C}_{\text{before}}$ vs. the more inclusive $\mathcal{C}_{\text{prepend}}$. As expected, considering only uphill paths leads to higher precision overall, as we avoid the case of ASes that uselessly tag their provider p's action communities on an AS-path that has already traversed p; the drawback is lower recall as less information is available for inferences. The figures also show the results when using $\mathcal{C}_{\text{prepend}} \bigcup \mathcal{C}_{\text{prefix\_tree}}$; overall, we find that the prefix tree nearly doubles the recall, at the cost of some loss of precision.

Figure 5.8 also shows that setting F to zero is too conservative. With this configuration, our algorithm infers few information communities as action communities, achieving very low recall. Very low thresholds perform best, as they allow for some noise (*i.e.*, action communities appearing with their ASes) and significantly improve recall without sacrificing precision. After this initial filtering (increasing F from zero

ingly large [13, 14]. Our use of the F1-score focuses on the worse-performing minority class (the positive inferences) and is thus a more relevant, *conservative* result.



(a) Precision

(b) Recall

Figure 5.9: Impact of varying the minimum number of Vantage Points (VPs) observing a community in Algorithm 2 (N^c_{vps}). We analyze the precision and recall from 2018 to 2023, showing stable performance for all datasets.

to, *e.g.,* 0.01), the performance of our algorithm is stable across all threshold values. Considering this finding, in the rest of this chapter we set $F = 0.01$.

We also require a minimum visibility of a BGP community at vantage points (VPs) to make inferences. If we increase the number of VPs where a community must be observed, the precision increases but recall decreases as we make fewer inferences. Figure 5.9 shows the precision and recall achievable when we compare the inference of action communities using the first RIB of December from 2018 to 2023, varying the number of vantage points (VPs). We note that the algorithm's performance as a function of configuration parameters is consistent, meaning that the algorithm's configuration does not need to be reevaluated often. Considering the inflection points in the graphs, we choose 3 VPs as the minimum for action community inference as a good trade-off between precision and recall. Different applications can increase the number of VPs if they benefit from higher precision, or decrease to favor recall.

Finally, a VP observing very few routes with a community could lead to incorrect inferences. Therefore, we also evaluate how many routes with a particular community a VP must have before we consider that (VP, community) in our inference.

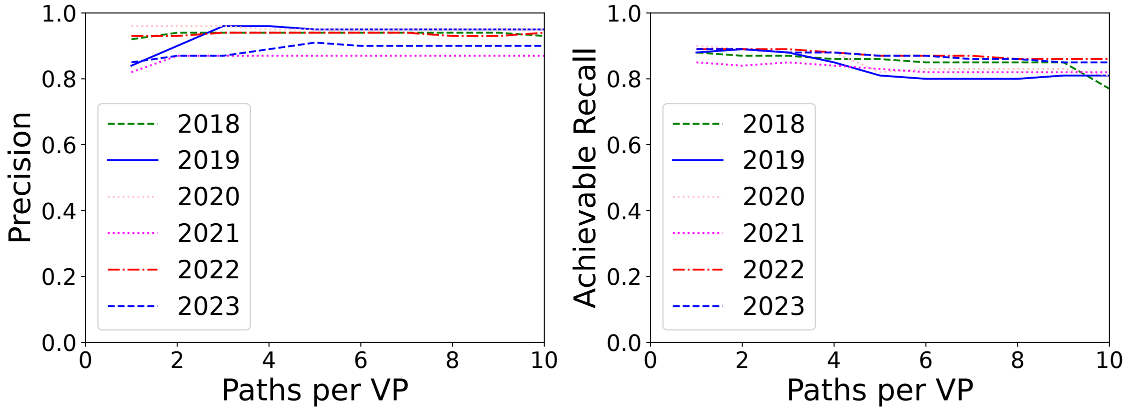


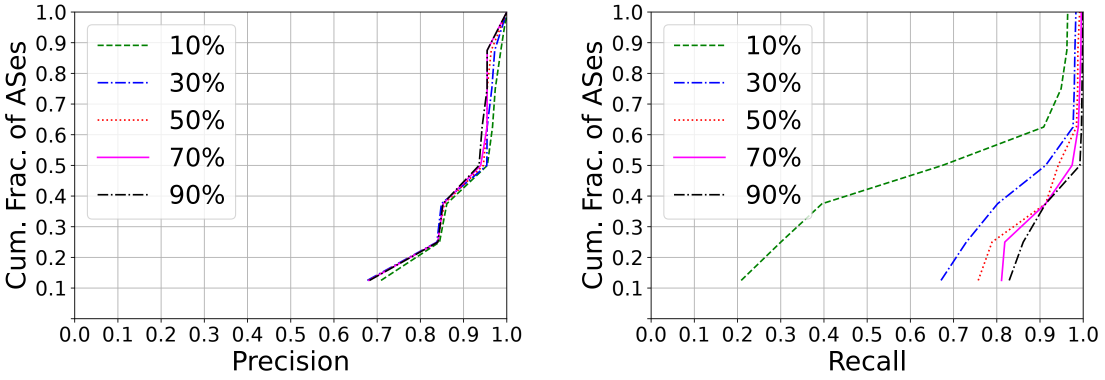
Figure 5.10: Performance as a function of the minimum number of required routes per (VP, community) pair before making inferences. Results are stable across the evaluation period. We conservatively chose a minimum of 4 routes per VP.

Figure 5.10 shows the impact of the minimum number of routes required when we fix $F = 0.01$ and the minimum number of VPs at 3, for every month of December between 2018 and 2023. Again, we observe that performance is stable throughout the period. We also find that the minimum number of required routes has limited impact, but that setting it too low may hurt precision. We take a conservative approach and set the minimum number of routes to 4 in the rest of the chapter, which the graphs indicate should work in general.

Building the Prefix-Tree

Section 5.3.3 proposed using a prefix tree for classifying communities. We evaluate how practical this approach is by evaluating how many communities are needed to build a prefix tree that achieves high precision and recall.

Figure 5.11a shows the distributions of precision and recall for 8 ASes with at least 20 communities in our ground-truth. We built prefix trees using action communities inferred with $\mathcal{C}_{\text{absent}}$, which avoids the loss of accuracy incurred by $\mathcal{C}_{\text{prepend}}$. Each point in the distribution represents the average of 100 executions with random subsets of the communities in $\mathcal{C}_{\text{absent}}$. The different lines vary the fraction of inferred communities used to build the tree. We report precision and recall obtained when classifying the communities in our ground-truth dataset using



(a) Precision (b) Recall

Figure 5.11: Cumulative distributions of precision and recall for inferences made by the prefix trees built from a random subset of inferred action communities. Different lines vary the fraction of inferences used as input to build the prefix tree and show that the prefix tree does not require many inferences to achieve high precision and recall.

the prefix tree.

We can see that prefix trees for most ASes achieve very high precision even when we build trees with as few as 10% of an AS's inferred action communities. Consequently, we need to infer only a small number of action communities for the prefix tree to be effective. Figure 5.11b shows that the recall is also high, increasing from an average of 0.66 when using 10% of the inferred communities to 0.95 when using 90%. For three of the 8 ASes, the recall is smaller than 0.4 for samples with 10% of the inferred communities, but it increases significantly for samples with 30% or more.

5.5.2 Inference Accuracy

Table 5.1 shows the number of communities, the *precision*, and *recall* for every Tier-1 and Tier-2 AS in our ground-truth dataset. The unknown columns (Unk) show the number of inferred action communities that are not in our ground-truth dataset. We color values larger than 0.8 green and values between 0.5 and 0.8 orange. We show three configurations of our algorithm: the baseline inferences ($\mathcal{C}_{\text{absent}}$), the inferences considering prepended paths ($\mathcal{C}_{\text{prepend}}$), and the inferences considering prepended

Table 5.1: Evaluation for ASes Tier-1 and Tier-2 on the BGP dumps from December 2023. The table shows the number of inferred communities (Num), precision (Prec), recall (Rec), and the number of inferred communities not in our ground-truth dataset (Unk) for three configurations of our algorithm. The last three columns (GT ∩ BGP) classify all communities appearing in the BGP dumps using our ground-truth dataset. The line Total at the bottom shows weighted averages of precision and recall.

	Inf. without Prep				**Inf. with Prep**				**Inf. Prep with Three**				**GT ∩ BGP**		
	$\mathcal{C}_{\text{absent}}$				$\mathcal{C}_{\text{prepend}}$				$\mathcal{C}_{\text{prepend}} \cup \mathcal{C}_{\text{tree}}$						
ASN	Num	Prec	Rec	Unk	Num	Prec	Rec	Unk	Num	Prec	Rec	Unk	Act	Info	Unk
1299	131	0.98	0.43	36	131	0.98	0.43	36	340	0.84	1.0	80	218	98	138
174	75	1.0	0.97	47	75	1.0	0.97	47	82	1.0	1.0	47	29	4	118
1764	2	0	0	2	13	1.0	0.18	6	13	1.0	0.18	6	38	38	16
2914	61	1.0	0.93	22	61	1.0	0.93	22	67	0.95	1.0	23	42	81	30
3257	36	0.88	0.38	19	36	0.88	0.38	19	61	0.87	0.85	23	39	844	26
3292	14	1.0	0.53	5	18	1.0	0.76	5	21	1.0	0.88	6	17	10	40
3356	27	0.75	0.6	23	37	0.5	0.6	31	239	0.38	0.6	231	5	144	331
33891	4	1.0	0.02	3	5	1.0	0.03	3	5	1.0	0.03	3	63	24	236
3491	62	0.94	0.25	12	69	0.95	0.28	13	252	0.94	0.99	50	16	139	60
3549	34	1.0	0.57	22	34	1.0	0.57	22	49	1.0	0.86	31	21	22	109
4589	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0
5400	2	0	0	0	2	0	0	0	2	0	0	0	1	0	68
5511	18	0.86	0.35	4	18	0.86	0.35	4	39	0.94	0.88	7	35	51	253
6461	43	1.0	0.64	4	43	1.0	0.64	4	63	0.95	0.92	4	61	289	54
6663	2	1.0	1.0	1	6	1.0	1.0	5	6	1.0	1.0	5	1	0	23
6762	65	1.0	0.13	54	69	0.93	0.16	54	196	0.68	1.0	70	86	46	79
701	19	1.0	1.0	12	19	1.0	1.0	12	20	1.0	1.0	13	7	0	16
7922	5	1.0	0.83	0	5	1.0	0.83	0	5	1.0	0.83	0	6	0	35
Total	600	0.97	0.37	266	641	0.96	0.40	283	1460	0.87	0.86	599	862	1794	1632

paths and the prefix tree ($\mathcal{C}_{\text{prepend}} \cup \mathcal{C}_{\text{tree}}$). As expected, relaxing the algorithm improves recall at the cost of precision. However, considering the small reduction in precision and large improvements in recall, we recommend the use of the inferences considering the prepended paths and the prefix tree. Applications where precision is paramount, however, can still opt for the more conservative configuration for the highest precision.

The last column (GT ∩ BGP) classifies the communities observed in the BGP dumps into action communities, information communities, or unknown depending on their type in our ground-truth dataset. This column shows that our algorithm achieves high precision and recall for the majority of ASes whose communities have

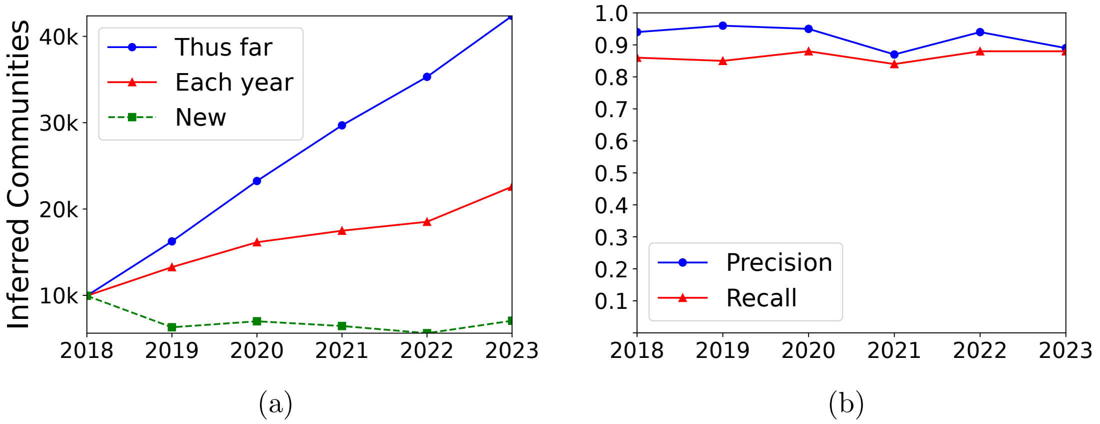


Figure 5.12: Results of our algorithms over six years (2018-2023): (a) the number of distinct communities found each year and (b) the precision and recall achieved. The data is based on the first BGP RIB collected in December from all RIPE and RouteViews collectors.

a significant presence in the BGP dumps. Our algorithm makes few inferences for ASes 4589 and 5400, which make limited use of BGP communities.

We carried out a longitudinal evaluation considering the first RIB of the month of December between 2018 and 2023. Figure 5.12 shows the results. On average, the precision is 92.5% (standard deviation of 3.62%) and the recall is 86.5% (standard deviation of 1.76%). Table 5.3 shows the type of correctly-inferred action communities across each year and demonstrates balanced semantic coverage across all action community classes. We classified the semantics of all but 1,617 action communities in our ground-truth dataset into the four classes in Table 5.3. For these communities we have no information to classify them, *e.g.*, 51 communities from AS5511 are labeled simply "tune" in the WHOIS documentation. Although we consider these action communities and correctly infer them as such, we do not include them in the table.

5.5.3 Clustering vs. Prefix Tree

Krenc *et al.* recently presented a mechanism for classifying BGP communities as action versus information [60]. They classify as action communities any community

Table 5.2: Comparison of cluster and prefix tree inferences for May 2023 by ASN that was not in the GT of the cluster inference algorithm. We compute the inference of the first RIB from all available collectors for the same interval used in [60] (who used all RIBs and updates). To be able to fully compare the communities captured across all collectors, we relaxed our algorithm to restrict to just one announcement per VP, while maintaining all other algorithm parameters. This way, both algorithms have the same visibility of all RIB communities. We cannot compute the Phi coefficient [18,66] when there are no inferences (positives) or when an AS has no documented information communities (true negatives).

	Prefix Tree					**GT vs BGP 7 days (1st RIB)**			**Cluster**					**GT vs BGP 7 days (all)**		
ASN	Infer	Prec	Rec	F1 Sc.	Phi	Act	Info	Unk	Infer	Prec	Rec	F1 Sc.	Phi	Act	Info	Unk
701	16	1.0	0.71	0.83	—	7	0	17	24	1.0	1.0	1.0	—	9	0	43
703	3	1.0	1.0	1.0	—	2	0	1	3	1.0	1.0	1.0	—	2	0	1
1764	22	1.0	0.41	0.58	0.50	41	36	14	0	0	0	0	—	41	40	14
3257	50	0.88	0.79	0.83	0.83	38	860	25	30	0.6	0.23	0.33	0.36	39	911	26
3549	55	1.0	0.91	0.95	0.91	22	22	129	63	1.0	0.3	0.47	0.42	23	22	144
4589	0	0	0	0	—	4	4	0	8	0.5	1.0	0.67	—	4	4	0
5400	1	0	0	0	—	2	0	62	3	1.0	1.0	1.0	—	2	0	62
5511	38	0.94	0.86	0.9	0.84	35	51	32	39	1.0	0.72	0.84	0.78	36	52	32
6663	12	1.0	1.0	1.0	—	1	0	15	7	1.0	1.0	1.0	—	1	0	17
7922	3	1.0	0.43	0.6	—	7	0	39	34	1.0	1.0	1.0	—	7	0	39
33891	55	1.0	0.7	0.82	0.63	63	24	245	35	1.0	0.19	0.32	0.25	64	24	247
Total	255	**0.96**	**0.68**	**0.80**	**0.78**	222	997	579	246	**0.89**	**0.35**	**0.50**	**0.51**	228	1053	625

that often appears on AS-paths that do not traverse the controlling AS.[3] The technique then clusters communities with integer values less than 140 apart and applies a majority vote across all communities in a cluster to determine their type. It reclassifies the communities in the minority group to match the type of the majority. The chapter evaluates the mechanism using ground truth from the NLNog database [73] and communities classified based on their descriptions using regular expressions.

We compared the inferences from our algorithm with the results available in their paper for the period they considered (May 1–7, 2023). We consider their original and our extended ground-truth datasets. On their ground-truth dataset, the prior work achieves an F1 score of 0.95 for the action communities, while our technique achieves 0.94. On our extended ground-truth dataset, the prior work achieves an F1 score of 0.92, while our technique also achieves 0.92.

However, there is a significant difference in performance when we consider only

[3] The specific threshold they use is 99.37% (a ratio of 160:1), which maximizes the F1 score for their ground-truth.

Table 5.3: Longitudinal Evaluation of Inferred Action Communities by Semantics.

	2018			2019			2020			2021			2022			2023		
Semantics	$\|\mathcal{C}\|$	BGP	Frac	$\|\mathcal{C}\|$	BGP	Frac	$\|\mathcal{C}\|$	BGP	Frac	$\|\mathcal{C}\|$	BGP	Frac	$\|\mathcal{C}\|$	BGP	Frac	$\|\mathcal{C}\|$	BGP	Frac
Local Preference	36	48	0.75	40	55	0.72	43	57	0.75	52	59	0.88	59	66	0.89	78	82	0.95
No Advertise/Export	145	165	0.88	168	169	0.99	179	183	0.98	187	226	0.83	221	222	0.99	207	234	0.88
Prepend (1x, 2x, 3x)	368	414	0.89	389	443	0.88	402	433	0.93	413	438	0.94	436	474	0.92	461	505	0.91
Blackhole	8	12	0.67	8	10	0.8	9	13	0.69	11	14	0.79	7	10	0.7	7	10	0.7

communities not in the original ground-truth dataset. Our algorithm achieves an F1 score of 0.8 vs. 0.5 for the previous technique, and Phi coefficient of 0.78 vs. 0.51. Table 5.2 shows a detailed evaluation of the Tier-1 and Tier-2 ASes present in our extended ground-truth but missing from the ground-truth dataset used by the prior work. These differences may be explained by the more recent publication of the communities of the ASes in the subset, which affects their visibility as fewer networks use them.

Also, some of these ASes assign community numbers in ways that violate the assumptions of the prior approach. The prior approach performs poorly for AS1764 (NextLayer) because AS1764 intermixes action and information communities when assigning community numbers, violating the assumption that ASes allocate communities in contiguous blocks. Table 5.4 shows how AS1764 groups information and action communities by neighbor, which leads to systematic errors when the majority vote is applied to communities in each cluster.

This behavior is not exclusive to AS1764; *e.g.,* AS3382 also groups communities by neighbor. This practice reduces the recall of our prefix tree, but there is no impact on precision as we do not overwrite inferences. AS3549 and AS33891 define both action and information communities in intervals smaller than 140, leading to low performance for the prior approach. Our approach performs better for these ASes, as the prefix tree can dynamically adjust group sizes.

Finally, the prior approach's worse performance for AS3257 (GTT) results from it not handling ASes squatting GTT's communities. We find that AS286 (previously KPN, acquired by GTT) and AS29140 (HostServer, unclear relationship to GTT)

Table 5.4: Example of AS1764's and AS33823's grouping of BGP communities by neighbor

Community	**Meaning**	**Category**
1764:40020	Received via Cogent AS174	Information
1764:40021	Prepend (1x) to Cogent AS174	Action
1764:40022	Prepend (2x) to Cogent AS174	Action
1764:40023	Prepend (3x) to Cogent AS174	Action
33823:1000	Announce International (default)	Information
33823:1001	Prepend (1x)	Action
33823:1002	Prepend (2x)	Action
33823:1003	Prepend (3x)	Action

both squat GTT's communities, leading the previous work to incorrectly infer some of GTT's information communities as action communities because they appear on routes without AS3257 (but with AS286 or AS29140). The similar F1 scores for the extended ground-truth dataset indicate that both techniques have similar overall performance, but our technique is more resilient to ASes with unknown operational practices or BGP community squatting.

5.6 Summary

In this chapter, we design and evaluate an algorithm for automatically identifying BGP action communities that relies only on route announcements observed by BGP route collectors. We also present an algorithm for uncovering ASes that consistently use (*i.e.,* squat) other ASes' communities, revealing undocumented relationships and shedding light on the complex interactions between networks on the Internet. These relationships help, for instance, filter out information communities that would otherwise be identified as action communities. Our evaluation results show that our algorithm for identifying action communities achieves average precision and recall of 92.5% and 86.5%, respectively, in a longitudinal study with BGP data from 2018 to 2023.

Chapter 6

Conclusion

This chapter summarizes our research on automatically inferring the semantics of BGP communities. In Chapters 4 and 5, we introduced methods to identify location and action communities. Location communities, a specific type of informational community, help trace the path of routing announcements, while action communities affect how these announcements are handled. Both types of data provide valuable information for network operators and researchers aiming to better understand and manage Internet traffic.

We explored the current state of the use of the BGP community, showing how these communities significantly impact Internet operations. The results of this thesis enable researchers and network operators to use our inferred datasets for multiple purposes, such as identifying anomalies in traffic patterns, detecting failures at Internet Exchange Points (IXPs), detecting traffic engineering from route announcements, and improving the understanding of network dynamics.

Our research addresses the challenge of inferring the semantics of BGP communities, particularly for poorly documented communities. We proposed automated classification techniques that work well in the wild for a subset of community types.

Our algorithms performed well in identifying location communities, achieving a precision of 93% and a recall of 81% for major Internet providers (Tier-1 and Tier-2 ASes). Compared to CAIDA's manually built database, our method provided similar accuracy but identified a far greater number of communities.

Additionally, our work automatically infers action communities and identifies autonomous systems involved in community squatting. Analyzing data from December 2018 to 2023, our algorithm for identifying action communities achieves an average precision of 92.5% and an average recall of 86.5%, demonstrating the robustness of our approach over multiple periods. Community squatting occurs when an AS uses BGP communities originally defined by another AS. Our method effectively detects this behavior, revealing hidden relationships between ASes. It can also improve methods for detecting sibling relationships.

This thesis provides new insights into BGP communities, showing their potential as a tool for optimizing network management. More research could unlock their full potential, helping to create more comprehensive and reliable databases to document metadata crucial to understanding and improving Internet routing.

6.1 Limitations and Future Work

The method presented in Chapter 4 focuses on inferring location communities. However, it requires routing announcements from K_{origins} distinct origin ASes to avoid cases where an origin AS tags all its announcements with the same traffic engineering communities from AS T. If any AS in $\mathcal{B}$ applies the same community tag to all routes, our algorithm may incorrectly classify traffic engineering communities as location communities.

The algorithm may also falsely infer a location community when AS T tags all routes received from a neighbor with a relationship community (*e.g.,* peer, customer, or provider). Although this case may decrease precision, it is not a significant issue, as ASes typically define only a few relationship communities, as discussed in Section 4.3. Requiring the community to appear on routes from neighbors with different relationships could mitigate this issue, although at the cost of reduced recall. We plan to explore this trade-off in future work.

Our algorithm also faces challenges when intermediate ASes between the BGP collector and the target AS T remove communities from BGP announcements [59]. However, the large number of collectors on the Internet provides sufficient visibility from multiple vantage points, allowing us to achieve high recall even when some ASes remove communities from the route announcements.

Chapter 5 details our work on inferring action communities and identifying ASes involved in community squatting. Although our approach can detect a single AS squatter per announcement, it is currently limited in identifying multiple squatters. We believe that multiple squatters exist, but further refinement is needed to capture more than one in the same announcement confidently.

Regarding action communities, we avoid using prepend communities to build the prefix tree because it can sometimes incorrectly infer informational communities, affecting the accuracy of our results. Investigating how to better separate informational communities from action communities will be a key area to improve our method.

This thesis has contributes to understanding the semantics of BGP communities, developing methods to classify both location and action communities, and revealing hidden relationships such as AS squatting. However, several areas remain for further exploration.

One of the primary challenges is accurately determining the geographic locations associated with BGP communities, which is still an open question. Furthermore, not all semantic meanings of action communities have been fully explored. While our method to identify prepend communities yielded promising results, certain action communities, such as NO-ADVERTISE, BLACKHOLE, and SELECTIVE ADVERTISE, continue to pose semantic ambiguities.

Future work should aim to resolve these ambiguities, clarify the meaning of action communities, and establish a more accurate geographic mapping of BGP communities. These advances will be crucial for optimizing the use of BGP communities in Internet routing.

Bibliography

[1] AMX-IX. AMS-IX Route Servers. `https://www.ams-ix.net/ams/documentation/ams-ix-route-servers`, 2024. [Online; accessed 25-Aug-2024].

[2] R. Anwar, H. Niaz, D. Choffnes, Í. Cunha, P. Gill, and E. Katz-Bassett. Investigating Interdomain Routing Policies in the Wild. In *Proceedings of the 2015 Internet Measurement Conference*, pages 71–77, Tokyo, Japan, 2015. ACM.

[3] A. Arturi, E. Carisimo, and F. E. Bustamante. as2org+: Enriching AS-to-Organization Mappings with PeeringDB. In *International Conference on Passive and Active Network Measurement*, pages 400–428. Springer, 2023.

[4] H. Birge-Lee, M. Apostolaki, and J. Rexford. Global BGP Attacks that Evade Route Monitoring, 2024.

[5] H. Birge-Lee, L. Wang, J. Rexford, and P. Mittal. SICO: Surgical Interception Attacks by Manipulating BGP Communities. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 431–448, London, United Kingdom, 2019. ACM Press.

[6] Business Wire. GTT Completes Acquisition of KPN International. `https://www.businesswire.com/news/home/20191202005214/en/GTT-Completes-Acquisition-of-KPN-International`, 2019. [Online; accessed 25-Aug-2024].

[7] CAIDA. CAIDA's Geolocation Dataset. `https://www.caida.org/catalog/datasets/bgp-communities/`, 2021. [Online; accessed 25-Aug-2024].

[8] CAIDA. The CAIDA AS Relationships Dataset, 2022-12-01. `https://www.caida.org/catalog/datasets/as-relationships/`, 2022. [Online; accessed 25-Aug-2024].

[9] CAIDA. CAIDA's AS-Organization Dataset. `http://data.caida.org/datasets/as-organizations/`, 2024. [Online; accessed 25-Aug-2024].

[10] A. Carvalho, B. A. Silva Jr, C. A. Silva, and R. A. Ferreira. Abrindo a Caixa-Preta - Aplicando IA Explicável para Aprimorar a Detecção de Sequestros de Prefixo. In *Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. SBC, 2024.

[11] R. Chandra, P. Traina, and T. Li. BGP Communities Attribute. Technical report, RFC 1997, August, 1996.

[12] Z. Chen, Z. S. Bischof, C. Testart, and A. Dainotti. Improving the Inference of Sibling Autonomous Systems. In *International Conference on Passive and Active Network Measurement*, pages 345–372. Springer, 2023.

[13] D. Chicco and G. Jurman. The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC genomics*, 21:1–13, 2020.

[14] D. Chicco, N. Tötsch, and G. Jurman. The Matthews Correlation Coefficient (MCC) is More Reliable than Balanced Accuracy, Bookmaker Informedness, and Markedness in Two-Class Confusion Matrix Evaluation. *BioData mining*, 14:1–22, 2021.

[15] M. Chiesa, L. Cittadini, G. Di Battista, L. Vanbever, and S. Vissicchio. Using Routers to Build Logic Circuits: How Powerful is BGP? In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, 2013.

[16] Y.-C. Chiu, B. Schlinker, A. B. Radhakrishnan, E. Katz-Bassett, and R. Govindan. Are We One Hop Away from a Better Internet? In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, IMC '15, pages 523–529, Tokyo, Japan, 2015. ACM Press.

[17] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill. BGP Hijacking Classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*, pages 25–32, Paris, France, 2019. IEEE.

[18] H. Cramér. *Mathematical Methods of Statistics*, volume 26. Princeton University Press, 41 William St, Princeton, NJ 08540, 1999.

[19] A. Dhamdhere, D. D. Clark, A. Gamero-Garrido, M. Luckie, R. K. Mok, G. Akiwate, K. Gogia, V. Bajpai, A. C. Snoeren, and K. Claffy. Inferring Persistent Interdomain Congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, pages 1–15, Budapest, Hungary, 2018. ACM.

[20] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, K. Claffy, and G. Riley. AS Relationships: Inference and Validation. *ACM SIGCOMM Computer Communication Review*, 37(1):29–40, 2007.

[21] B. Donnet. Incentives for BGP Guided IP-Level Topology Discovery. In *International Workshop on Traffic Monitoring and Analysis*, pages 101–108, Springer Berlin Heidelberg, 2009. TMA'09, Springer.

[22] B. Donnet and O. Bonaventure. On BGP Communities. *ACM SIGCOMM Computer Communication Review*, 38(2):55–59, 2008.

[23] J. Durand, I. Pepelnjak, and G. Döring. BGP7454: BGP Operations and Security. Technical report, RFC 7454, February, 2015.

[24] N. Feamster, Z. M. Mao, and J. Rexford. BorderGuard: Detecting Cold Potatoes from Peers. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 213–218, 2004.

[25] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet Routing Instabilities. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, page 205–218, New York, NY, USA, 2004. ACM.

[26] G. Feng, S. Seshan, and P. Steenkiste. UNARI: an Uncertainty-Aware Approach to AS Relationships Inference. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, pages 272–284, 2019.

[27] O. Fonseca, Í. Cunha, E. Fazzion, B. Junior, R. A. Ferreira, and E. Katz-Bassett. Tracking Down Sources of Spoofed IP Packets. In *IFIP Networking Conference*, pages 51–53, 2019.

[28] O. Fonseca, Í. Cunha, E. Fazzion, W. Meira, B. A. da Silva, R. A. Ferreira, and E. Katz-Bassett. Identifying Networks Vulnerable to IP Spoofing. *IEEE Transactions on Network and Service Management*, 18(3):3170–3183, 2021.

[29] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, 2001.

[30] L. Gao and J. Rexford. Stable Internet Routing Without Global Coordination. *IEEE/ACM Transactions on Networking*, 9(6):12, 2001.

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

[32] GBLX. GBLX Customer BGP Communities. `https://onestep.net/communities/as3549/`, 2008. [Online; accessed 25-Aug-2024].

[33] P. Gill, M. Schapira, and S. Goldberg. Let the Market Drive Deployment: A Strategy for Transitioning to BGP Security. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, page 14–25, New York, NY, USA, 2011. Association for Computing Machinery.

[34] P. Gill, M. Schapira, and S. Goldberg. A Survey of Interdomain Routing Policies. *ACM SIGCOMM Computer Communication Review*, 44(1):28–34, 2013.

[35] V. Giotsas, C. Dietzel, G. Smaragdakis, A. Feldmann, A. Berger, and E. Aben. Detecting Peering Infrastructure Outages in the Wild. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 446–459, New York, NY, USA, 2017. ACM. event-place: Los Angeles, CA, USA.

[36] V. Giotsas, T. Koch, E. Fazzion, Í. Cunha, M. Calder, H. V. Madhyastha, and E. Katz-Bassett. Reduce, Reuse, Recycle: Repurposing Existing Measurements to Identify Stale Traceroutes. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, pages 247–265, New York, NY, USA, 2020. ACM Press.

[37] V. Giotsas, M. Luckie, B. Huffaker, and k. claffy. Inferring Complex AS Relationships. In *Proceedings of the ACM Internet Measurement Conference*, IMC '14, page 23–30, New York, NY, USA, 2014. ACM Press.

[38] V. Giotsas, G. Smaragdakis, C. Dietzel, P. Richter, A. Feldmann, and A. Berger. Inferring BGP Blackholing Activity in the Internet. In *Proceedings of the ACM Internet Measurement Conference*, IMC '17, pages 1–14, New York, NY, USA, 2017. ACM Press.

[39] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and k. claffy. Mapping Peering Interconnections to a Facility. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies - CoNEXT '15*, pages 1–13, Heidelberg, Germany, 2015. ACM Press.

[40] V. Giotsas, S. Zhou, M. Luckie, and k. claffy. Inferring Multilateral Peering. In *Proceedings of the 9th ACM conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, page 247–258, New York, NY, USA, 2013. ACM Press.

[41] E. Gregori, A. Improta, L. Lenzini, L. Rossi, and L. Sani. BGP and Inter-AS Economic Relationships. In *International Conference on Research in Networking*, pages 54–67, Valencia, Spain, 2011. Springer, Springer.

[42] E. Gregori, A. Improta, and L. Sani. Isolario: A Do-ut-des Approach to Improve the Appeal of BGP Route Collecting, 2016.

[43] J. Heitz, J. Snijders, K. Patel, I. Bagdonas, and N. Hilliard. RFC8092: BGP Large Communities Attribute, 2017.

[44] R. V. Hogg, J. McKean, and A. T. Craig. *Introduction to Mathematical Statistics*. Pearson Education, Upper Saddle River, N.J, 2005.

[45] House, Packet Clearing. Packet Clearing House. `https://www.pch.net/`, 2024. [Online; accessed 25-Aug-2024].

[46] G. Huston. NOPEER Community for Border Gateway Protocol (BGP) Route Scope Control. RFC 3765, Apr. 2004.

[47] C. Inc. CISCO Route-Map. `https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/49111-route-map-bestp.html`, 2024. [Online; accessed 25-Aug-2024].

[48] Internet Assigned Numbers Authority. Border Gateway Protocol (BGP) Well-known Communities. `https://www.iana.org/assignments/bgp-well-known-communities`, 2023. [Online; accessed 25-Aug-2024].

[49] Isolario Project. BGPScanner. `https://gitlab.com/Isolario/bgpscanner`. [Online; accessed 25-Aug-2024].

[50] Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy, and S. Shenker. Stable and practical AS relationship inference with ProbLink. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 581–598, Boston, MA, Feb. 2019. USENIX Association.

[51] B. Junior, R. A. Ferreira, Í. Cunha, B. Schlinker, and E. Katz-Bassett. High-Fidelity Interdomain Routing Experiments. In *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*, pages 36–38, 2018.

[52] B. A. S. Junior, A. B. de Carvalho, Ítalo Cunha, T. Friedman, E. Katz-Bassett, and R. A. Ferreira. BGP Action Communities – Supplemental Material. `https://github.com/TopoMapping/bgp-action-communities`, 2024.

[53] B. A. S. Junior, P. Mol, O. Fonseca, Ítalo Cunha, R. A. Ferreira, and E. Katz-Bassett. BGP Communities – Supplemental Material. `https://github.com/TopoMapping/bgp-communities`, 2021.

[54] R. M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[55] E. Katz-Bassett, C. Scott, D. R. Choffnes, Í. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. LIFEGUARD: Practical Repair of Persistent Route Failures. In *Proceedings of the 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '12, pages 395–406, New York, NY, USA, 2012. ACM Press.

[56] T. King, C. Dietzel, J. Snijders, G. Doering, and G. Hankins. BLACKHOLE Community. RFC 7999, Oct. 2016.

[57] M. Konte, R. Perdisci, and N. Feamster. ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 625–638, New York, NY, USA, 2015. ACM Press.

[58] T. Krenc, R. Beverly, and G. Smaragdakis. 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*, pages 443–450, New York, NY, USA, 2020. ACM Press.

[59] T. Krenc, R. Beverly, and G. Smaragdakis. AS-Level BGP Community Usage Classification. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, page 577–592, New York, NY, USA, 2021. Association for Computing Machinery.

[60] T. Krenc, M. Luckie, A. Marder, and k. claffy. Coarse-grained Inference of BGP Community Intent. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 66–72, 2023.

[61] W. Kumari and K. Sriram. RFC 6472-Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP, 2011.

[62] T. Li, R. Chandra, and P. S. Traina. BGP Communities Attribute. RFC 1997, Aug. 1996.

[63] Z. Li, D. Levin, N. Spring, and B. Bhattacharjee. Internet Anycast: Performance, Problems, & Potential. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, pages 59–73, Budapest, Hungary, 2018. ACM Press.

[64] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k. claffy. AS Relationships, Customer Cones, and Validation. In *Proceedings of the ACM Internet Measurement Conference*, IMC '13, pages 243–256, Barcelona, Spain, 2013. ACM Press.

[65] P. Marcos, L. Prehn, L. Leal, A. Dainotti, A. Feldmann, and M. Barcellos. AS-Path Prepending: There is no Rose without a Thorn. In *Proceedings of the ACM Internet Measurement Conference*, pages 506–520, 2020.

[66] B. W. Matthews. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[67] F. Mazzola, P. Marcos, and M. Barcellos. Light, Camera, Actions: Characterizing the Usage of IXPs' Action BGP Communities. In *Proceedings of*

the 18th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '22, page 196–203, New York, NY, USA, 2022. Association for Computing Machinery.

[68] D. Meyer. University of Oregon Route Views Archive Project. `http://www.routeviews.org/`, 1997. [Online; accessed 25-Aug-2024].

[69] R. Miller. Level 3 Buys Global Crossing for $3 Billion, 2011. `https://www.datacenterknowledge.com/archives/2011/04/11/level-3-buys-global-crossing-for-3-billion`.

[70] A. Milolidakis, T. Bühler, K. Wang, M. Chiesa, L. Vanbever, and S. Vissicchio. On the Effectiveness of BGP Hijackers That Evade Public Route Collectors. *IEEE Access*, 11:31092–31124, 2023.

[71] F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz, et al. Sustainable Data Analysis with Snakemake. *F1000Research*, 10:10–33, 2021.

[72] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-Topology Model that Captures Route Diversity. *ACM SIGCOMM Computer Communication Review*, 36(4):195–206, 2006.

[73] Netherlands Network Operator Group. NLNOG Looking Glass - Known communities. `https://github.com/NLNOG/lg.ring.nlnog.net/tree/main/communities`, 2023. [Online; accessed 25-Aug-2024].

[74] L. Prehn and A. Feldmann. How Biased is Our Validation (data) for AS Relationships? In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 612–620, 2021.

[75] B. Quoitin and O. Bonaventure. A Survey of the Utilization of the BGP Community Attribute. Internet-Draft draft-quoitin-bgp-comm-survey-00, Internet Engineering Task Force, Mar. 2002. Work in Progress.

[76] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig. Interdomain Traffic Engineering with BGP. *IEEE Communications magazine*, 41(5):122–128, 2003.

[77] B. Quoitin, S. Uhlig, and O. Bonaventure. 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*, QofIS'02/ICQT'02, page 125–134, Berlin, Heidelberg, 2002. Springer-Verlag.

[78] R. Raszuk, J. Haas, A. Lange, B. Decraene, S. Amante, and P. Jakma. BGP Community Container Attribute, 2023.

[79] Y. Rekhter. RFC 4271: A Border Gateway Protocol 4 (BGP-4), 2006.

[80] Y. Rekhter and T. Li. RFC 1654: A Border Gateway Protocol 4 (BGP-4), 1994.

[81] N. RIPE. RIPE RIS Project. `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/`, 2024. [Online; accessed 25-Aug-2024].

[82] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821, Oct. 2011.

[83] L. Salamatian, T. Arnold, Í. Cunha, J. Zhu, Y. Zhang, E. Katz-Bassett, and M. Calder. Who Squats IPv4 Addresses? *ACM SIGCOMM Computer Communication Review*, 53(1):48–72, 2023.

[84] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack. HEAP: Reliable Assessment of BGP Hijacking Attacks. *IEEE Journal on Selected Areas in Communications*, 34(6):1849–1861, 2016.

[85] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. 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*, SIGCOMM '17, page 418–431, New York, NY, USA, 2017. ACM Press.

[86] M. Sendra, R. Sutrisno, J. Harianata, D. Suhartono, and A. B. Asmani. Enhanced Latent Semantic Analysis by Considering Mistyped Words in Automated Essay Scoring. In *2016 International Conference on Informatics and Computing (ICIC)*, pages 304–308, 2016.

[87] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti. ARTEMIS: Neutralizing BGP Hijacking within a Minute. *IEEE/ACM Transactions on Networking*, 26(6):2471–2486, 2018.

[88] B. A. Silva Jr, A. Carvalho, I. Cunha, T. Friedman, E. Katz-Bassett, and R. A. Ferreira. Uncovering BGP Action Communities and Community Squatters in the Wild. *In Proceedings of ACM SIGMETRICS / IFIP Performance*, 2025.

[89] B. A. Silva Jr, P. Mol, O. Fonseca, I. Cunha, R. A. Ferreira, and E. Katz-Bassett. Automatic Inference of BGP Location Communities. *In Proceedings of ACM SIGMETRICS / IFIP Performance*, 6(1):1–23, 2022.

[90] P. Slavík. A Tight Analysis of the Greedy Algorithm for Set Cover. In *ACM STOC*, STOC '96, page 435–441, Philadelphia, Pennsylvania, USA, 1996. ACM.

[91] O. Step. One Step. `https://onestep.net/communities/`, 2015. [Online; accessed 25-Aug-2024].

[92] O. Step. AS3257 Public Information. `https://onestep.net/communities/as3257/`, 2021. [Online; accessed 25-Aug-2024].

[93] F. Streibelt, F. Lichtblau, R. Beverly, A. Feldmann, C. Pelsser, G. Smaragdakis, and R. Bush. BGP Communities: Even More Worms in the Routing

Can. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, page 279–292, New York, NY, USA, 2018. ACM.

[94] P. Sun, L. Vanbever, and J. Rexford. Scalable Programmable Inbound Traffic Engineering. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, pages 1–7, 2015.

[95] D. Tappan, S. R. Sangli, and Y. Rekhter. BGP Extended Communities Attribute, 2006.

[96] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of Hot-Potato Routing in IP Networks. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, pages 307–319, 2004.

[97] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark. Profiling BGP Serial Hijackers: Capturing Persistent Misbehavior in the Global Routing Table. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 420–434, New York, NY, USA, 2019. Association for Computing Machinery.

[98] A. Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 17(1):168–192, 2020.

[99] I. Tools. Whois Servers List. `https://www.mobilefish.com/tutorials/whois_servers_list/whois_servers_list.html`, 2024. [Online; accessed 25-Aug-2024].

[100] F. Wang and L. Gao. On Inferring and Characterizing Internet Routing Policies. *Journal of Communications and Networks*, 9(4):350–355, 2007.

[101] T. F. E. Wikipedia. Tier 1 Network. `https://en.wikipedia.org/wiki/Tier_1_network`, 2024. [Online; accessed 25-Aug-2024].

[102] T. F. E. Wikipedia. Tier 2 Network. `https://en.wikipedia.org/wiki/Tier_2_network`, 2024. [Online; accessed 25-Aug-2024].

[103] R. Wray and D. Milmo. Watchdog clears BSkyB acquisition of Easynet. The Guardian, 2005. `https://www.theguardian.com/technology/2005/dec/31/news.citynews`.

[104] J. Xia and L. Gao. On the Evaluation of AS Relationship Inferences [Internet Reachability/Traffic Flow Applications]. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, volume 3, pages 1373–1377 Vol.3, Dallas, TX, USA, 2004. IEEE.

[105] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, and A. Jain. 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*, pages 432–445, Los Angeles, CA, USA, 2017. ACM.

[106] M. Zeng, D. Li, P. Zhang, K. Xie, and X. Huang. Federated Route Leak Detection in Inter-domain Routing with Privacy Guarantee. *ACM Trans. Internet Technol.*, Feb 2023.

[107] Y. Zhang and M. Pourzandi. Studying Impacts of Prefix Interception Attack by Exploring BP AS-path Prepending. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pages 667–677. IEEE, 2012.

[108] Y. Zhang and M. Tatipamula. Characterization and Design of Effective BGP AS-path Prepending. In *2011 19th IEEE International Conference on Network Protocols*, pages 59–68, Dallas, TX, USA, 2011. IEEE.

Appendix A

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. We employ a non-invasive approach that does not disrupt Internet announcements, and all processing is performed offline.

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 [5, 93]. 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.

We also infer action communities and AS *squatters*. Actions communities can be used maliciously and cause damage to the Internet infrastructure [5, 93]. Since we only identify action communities and not their complete semantics (*e.g.,* no-export, prepend), a malicious actor would have to acquire additional information to launch an attack on a specific AS.

Our community and AS squatters databases 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 our public databases far outweigh the possibility of misuse for malicious activities.

Appendix B

Publications

We presented the results of Chapter 4 at ACM SIGMETRICS / IFIP PERFORMANCE, the flagship conference of the ACM special interest group for the computer systems performance evaluation community and of the IFIP working group WG7.3 on performance modeling and analysis. The results of Chapter 5 will be presented at ACM SIGMETRICS 2025. Additionally, we contributed to publications in other conferences and journals, such as IFIP Networking, IEEE Transactions on Network and Service Management (TNSM), ACM CoNEXT, and ACM SIGCOMM. We produced the following manuscripts during the development of this thesis:

- **SILVA JR. B. A.**; MOL, P.; FONSECA, O.; CUNHA, I.; FERREIRA, R. A.; and KATZ-BASSETT, E. Automatic Inference of BGP Location Communities. Published at the ACM SIGMETRICS / IFIP PERFORMANCE, 2022. [89]
- **SILVA JR. B. A.**; CARVALHO, A.; CUNHA, I.; FRIEDMAN, T.; KATZ-BASSETT, E.; and FERREIRA, R. A. Uncovering BGP Action Communities and Community Squatters in the Wild. To appear at ACM SIGMETRICS, 2025. [88]
- **SILVA JR, B. A.**; FERREIRA, R. A.; CUNHA, I. F. S.; SCHLINKER, B.; and KATZ-BASSET, E. Poster: High-Fidelity Interdomain Routing Experiments. In Proceedings of the ACM SIGCOMM Posters and Demos, pages 36-38, Budapest, Hungary, August 20-25, 2018. [51]

- FONSECA, O.; CUNHA I.; FAZZION, E.; MEIRA, W.; **SILVA JR, B. A.**; FERREIRA, R. A.; and KATZ-BASSET, E. Identifying Networks Vulnerable to IP Spoofing. IEEE Transactions on Network and Service Management (IEEE TNSM), 2021. p. 3170-3183. [28]

- FONSECA, O.; CUNHA, I. F. S.; FAZZION, E.; MEIRA, W.; **SILVA JR, B. A.**; FERREIRA, R. A.; and KATZ-BASSETT, E. Tracking Down Sources of Spoofed IP Packets. In: IFIP Networking Conference (NETWORKING), 2020, Paris, France. IFIP Networking Conference, 2020. p. 208-216 (Awarded Best-Paper Runner-up). [27]

- CARVALHO, A.; **SILVA JR, B. A.**; SILVA, C. A.; and FERREIRA, R. A. Abrindo a Caixa-Preta - Aplicando IA Explicável para Aprimorar a Detecção de Sequestros de Prefixo. Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg), 2024. [10]

- FONSECA, O.; CUNHA, I.; FAZZION, E.; **JUNIOR, B.**; FERREIRA, R. A.; and KATZ-BASSET, E. Tracking Down Sources of Spoofed IP Packets. In Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies (Poster), December, 2019. [27]

Although not directly related to the core topic of this thesis, our previous work on routing experiments [51] and detection of spoofed IP packets [27, 28] provided valuable experience in processing large BGP data dumps, consolidating data from diverse sources, analyzing algorithmic outputs, and conducting experiments on the PEERING testbed. The paper [10] helped me initiate the mentoring of master's students, which is an invaluable skill for a researcher. In summary, these publications contributed significantly to the development of this thesis and my education.