How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric

Bahador Yeganeh, Ramakrishnan Durairajan,
Reza Rejaie
{byeganeh,ram,reza}@cs.uoregon.edu
University of Oregon

Walter Willinger
wwillinger@niksun.com
NIKSUN, Inc.

ABSTRACT

The growing demand for an ever-increasing number of cloud services is profoundly transforming the Internet's interconnection or peering ecosystem, and one example is the emergence of "virtual private interconnections (VPIs)". However, due to the underlying technologies, these VPIs are not publicly visible and traffic traversing them remains largely hidden as it bypasses the public Internet. In particular, existing techniques for inferring Internet interconnections are unable to detect these VPIs and are also incapable of mapping them to the physical facility or geographic region where they are established.

In this paper, we present a third-party measurement study aimed at revealing all the peerings between Amazon and the rest of the Internet. We describe our technique for inferring these peering links and pay special attention to inferring the VPIs associated with this largest cloud provider. We also present and evaluate a new method for pinning (i.e., geo-locating) each end of the inferred interconnections or peering links. Our study provides a first look at Amazon's peering fabric. In particular, by grouping Amazon's peerings based on their key features, we illustrate the specific role that each group plays in how Amazon peers with other networks.

CCS CONCEPTS

• **Networks → Routers**; **Network measurement**; **Physical topologies**; **Logical / virtual topologies**; *Cloud computing*;

1 INTRODUCTION

A myriad of new cloud service offerings made possible by modern-day cloud computing is fundamentally changing how business is conducted in all segments of the private and public sectors. This, in turn, has transformed the way these companies connect to major cloud service providers to utilize these services. In particular, many companies prefer to bypass the public Internet and directly connect to major cloud service providers at a close-by colocation (or colo) facility to experience better performance when using these cloud services. In response to these demands, some of the major colo facilities have started to deploy and operate new switching infrastructure called *cloud exchanges* [23, 25]. Importantly, in conjunction with this new infrastructure, these colo providers have also introduced a new interconnection service offering called *"virtual private interconnection (VPI)"* [3, 43, 58]. By purchasing a single port on the cloud exchange switching fabric in a given facility, VPIs enable enterprises that are either natively deployed in that facility or "brought" into the facility by their upstream providers to establish direct peering to any number of cloud service providers that are present on that exchange.

The implications of this transformation for the Internet's interconnection ecosystem have been profound. First, the on-demand nature of VPIs introduces a degree of dynamism into the Internet interconnection fabric that has been missing in the past where setting up traditional interconnections of the public or private peering types took days or weeks. Second, once the growing volume of an enterprise's traffic enters an existing VPI to a cloud provider, it is handled entirely by that cloud provider's private infrastructure (i.e., the cloud provider's private backbone that interconnects its own datacenters) and completely bypasses the public Internet. Finally, none of these VPIs are visible to existing methods and tools that have been specifically designed to infer and/or map the interconnections in today's Internet [2, 55, 57, 62].

Among the reasons for this shortcoming of the existing inference or mapping tools is the fact that, due to their traceroute-based nature and their reliance on conventional measurement platforms, they lack cloud-centric vantage points (e.g., Virtual Machines (VMs) running in Amazon AWS). A second and more important reason is that the existing techniques for inferring interconnections are, in general, incapable of revealing the connectivity at the newly emerging switching fabrics (e.g., cloud exchanges), mainly because of these fabrics' reliance on layer-2 technologies. In short, from an Internet measurement perspective, not only are VPIs by and large invisible to existing methods for Internet connectivity discovery, but any traffic traversing these VPIs is only visible to the corresponding cloud provider and can therefore no longer be accounted for by traditional traffic monitoring or traffic estimation efforts.

This paper's main contribution consists of presenting a third-party, cloud-centric measurement study aimed at discovering and characterizing the unique peerings (along with their types) of Amazon, the largest cloud service provider in the US and worldwide. Each peering typically consists of one or multiple (unique) interconnections between Amazon and a neighboring Autonomous System (AS) that are typically established at different colocation facilities around the globe. Our study only utilizes publicly available information and data (i.e., no Amazon-proprietary data is used) and is

IMC '19, October 21–23, 2019, Amsterdam, Netherlands
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6948-0/19/10...$15.00
https://doi.org/10.1145/3355369.3355602

therefore also applicable for discovering the peerings of other large cloud providers.[1]

We start by presenting the required background on Amazon's serving infrastructure, including the different types of peerings an enterprise network can establish with Amazon at a colo facility. We also provide a summary of prior work in this area in § 2. § 3 describes the first round of our data collection; that is, launching cloud-centric traceroute probes from different regions of Amazon's infrastructure toward all the /24 (IPv4) prefixes to infer a subset of Amazon's peerings. We present our methodology for inferring Amazon's peerings across the captured traceroutes in § 4.1. Our second round of data collection consists of using traceroute probes that target the prefixes around the peerings discovered in the first round and are intended to identify all the remaining (IPv4) peerings of Amazon (§ 4.2). In § 5, we present a number of heuristics to resolve the inherent ambiguity in inferring the specific traceroute segment that is associated with a peering. We further confirm our inferred peerings by assessing the consistency of border interfaces at both the Amazon side and client side of an inferred interconnection.

Pinning (or geo-locating) each end of individual interconnections associated with Amazon's peerings at the metro level forms another contribution of this study (§ 6). To this end, we develop a number of methods to identify border interfaces that have a reliable location and which we refer to as anchors. Next, we establish a set of co-presence rules to conservatively propagate the location of anchors to other close-by interfaces. We then identify the main factors that limit our ability to pin all border interfaces at the metro level and present ways to pin most of the interfaces at the regional level. Finally, we evaluate the accuracy and coverage of our pinning technique and characterize the pinned interconnections.

The final contribution of this paper is a new method for inferring the client border interface that is associated with that client's VPI with Amazon. In particular, by examining the reachability of a given client border interface from a number of other cloud providers (§ 7) and identifying overlapping interfaces between Amazon and those other cloud providers, our method provides a lower bound on the number of Amazon's VPIs. We then assign all inferred Amazon peerings to different groups based on their key attributes such as being public or private, visible or not visible in BGP, and physical or virtual. We then carefully examine these groups of peerings to infer their purpose and explore hybrid peering scenarios. In particular, we show that one-third of Amazon's inferred peerings are either virtual or not visible in BGP and thus hidden from public measurement. Finally, we characterize the inferred Amazon connectivity graph as a whole.

Overall, our analysis of Amazon's peering fabric highlights how (e.g., using virtual and non-BGP peerings) and where (e.g., at which metro) Amazon's cloud traffic "goes hiding"; that is, bypasses the public Internet. In particular, we show that as large cloud providers such as Amazon aggressively pursue new connect locations closer to the Internet's edge, VPIs are an attractive interconnection option as they *(i)* create shortcuts between enterprises at the edge of the network and the large cloud providers (i.e., further contributing to the flattening of the Internet) and *(ii)* ensure that cloud-related traffic is primarily carried over the large cloud providers' private backbones (i.e., not exposed to the unpredictability of the best-effort public Internet).

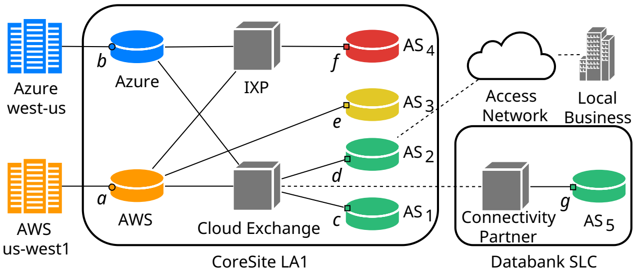


Figure 1: Overview of Amazon's peering fabric. Native routers of Amazon & Microsoft (orange & blue) establishing private interconnections (AS_3 - yellow router), public peering through IXP switch (AS_4 - red router), and virtual private interconnections through cloud exchange switch (AS_1, AS_2, and AS_5 - green routers) with other networks. Remote peering (AS_5) as well as connectivity to non-ASN businesses through layer-2 tunnels (dashed lines) happens through connectivity partners.

2 BACKGROUND AND RELATED WORK

Amazon's Ecosystem. The focus of our study of peerings in today's Internet is Amazon, arguably the largest cloud service provider in the US and worldwide. Amazon operates several data centers worldwide. While these data centers' street addresses are not *explicitly* published by Amazon, their geographic locations have been reported elsewhere [14, 24, 60, 68, 79, 80]. Each data center hosts a large number of Amazon servers that, in turn, host user VMs as well as other services (e.g., Lambda). Amazon's data center locations are divided into independent and distinct geographic *regions* to achieve fault tolerance/stability. Specifically, each region has multiple, isolated *availability zones (AZs)* that provide redundancy and offer high availability in case of failures. *AZs* are virtual and their mapping to a specific location within their region is not known [8]. As of 2018, Amazon had 18 regions (55 *AZs*) across the world, with five of them (four public + one US government cloud) located in the US. For our study, we were not able to utilize three of these regions. Two of them are located in China, are not offered on Amazon's AWS portal, and require approval requests by Amazon staff. The third region is assigned to the US government and is not offered to the public.

Peering with Amazon at Colo Facilities. Clients can connect to Amazon through a specific set of colo facilities. Amazon is considered a *native* tenant in these facilities, and their locations are publicly announced by Amazon [4]. Amazon is also *reachable* through a number of other colo facilities via layer-2 connectivity offered by third-party providers (e.g., Megaport).[2]

Figure 1 depicts an example of different types of peerings offered by cloud providers at two colo facilities. Both Amazon (AWS) and Microsoft (Azure) are native (i.e., house their border routers) in the CoreSite LA1 colo facility and are both present at that facility's IXP

[1] As long as the cloud provider does not filter traceroute probes.

[2] These entities are called "AWS Direct Connect Partners" at a particular facility and are listed online along with their points of presence [6].

and cloud exchange. (Open) cloud exchanges are switching fabrics specifically designed to facilitate interconnections among network providers, cloud providers, and enterprises in ways that provide the scalability and elasticity essential for cloud-based services and applications (e.g., see [23, 34]). Major colo facility providers (e.g., Equinix and CoreSite) also offer a new interconnection service option called "virtual private interconnection (VPI)." VPIs enable local enterprises (that may or may not own an ASN) to connect to multiple cloud providers that are present at the cloud exchange switching fabric by means of purchasing a single port on that switch. In addition, VPIs provide their customers access to a programmable, real-time cloud interconnection management portal. Through this portal, the operators of these new switching fabrics make it possible for individual enterprises to establish their VPIs in a highly-flexible, on-demand, and near real-time manner. This portal also enables enterprises to monitor in real-time the performance of their cloud-related traffic that traverses these VPIs.

While cloud exchanges rely on switching fabrics that are similar to those used by IXPs, there are two important differences. For one, cloud exchanges enable each customer to establish virtualized peerings with multiple cloud providers through a single port. Moreover, they provide exclusive client connectivity to cloud providers without requiring a client to use its pre-allocated IP addresses. Operationally, a cloud customer establishes VPIs using either public or private IP addresses depending on the set of cloud services that this customer is trying to reach through these interconnections. On the one hand, VPIs relying on private addresses are limited to the customer's virtual private cloud (VPC) through VLAN isolation. On the other hand, VPIs with public addresses can reach compute resources in addition to other AWS offerings such as S3 and DynamoDB [5]. Given the isolation of network paths for VPIs with private addresses, any peerings associated with these VPIs are not visible to the probes from VMs owned by other Amazon customers. This makes it, in practice, impossible to discover established VPIs that rely on private addresses. In Figure 1, the different colors of the client routers indicate the type of their peerings; e.g., public peering through the IXP (for AS_4), direct physical interconnection (also called "cross-connect") (for AS_3), private virtual peerings that are either local (for AS_1 and AS_2) or remote (for AS_5). Here, a local virtual private peering (e.g., AS_2) could be associated with an enterprise that is brought to the cloud exchange by its access network (e.g., Comcast) using layer-2 technology; based on traceroute measurements, such a peering would appear to be between Amazon and the access network. In contrast, a remote private virtual peering could be established by an enterprise (e.g., AS_5) that is present at a colo facility (e.g., Databank in Salt Lake City in Figure 1) where Amazon is not native but that houses an "AWS Direct Connect Partner" (e.g., Megaport) which in turn provides layer-2 connectivity to AWS.

Related Work. Discovering the AS-level topology of the Internet has been of interest to the networking community for decades [20, 27, 29, 30, 40, 51, 52, 70]. Another body of work focuses on providing a physical map of the Internet infrastructure (e.g., colocation facilities, fiber-optic cables). Commonly-used techniques in this domain include parsing of DNS names to extract details about infrastructure and geography [19, 46, 61, 66, 75–77], performing extensive web searches [31, 33], relying on geolocation information [35, 36, 45, 48, 65, 69, 78, 82], and leveraging RTT-based estimation techniques [17, 18, 40].

Recent studies have limited their focus to identifying interconnections and sub-structures of the Internet such as identifying POPs [9, 76], elucidating public peerings at Internet Exchange Points (IXPs) [1, 10, 21, 42, 71, 83], enhancing connectivity discovery using hybrid approaches [32], and identifying interconnections [51, 55, 57]. Other studies in this area focus on how IXPs are reshaping the Internet's AS-level topology from a pronounced hierarchical construct to a more mesh-like network [26, 39]. Yet other efforts are expanded to develop different alias resolution techniques to enhance the accuracy of inferred router-level topologies [12, 49, 50, 73, 74]. Our work is complementary to these studies and describes the yet largely unknown contributions of the largest cloud providers to the connectivity fabric of today's Internet.

Our work is closely related to recent studies that concern the serving infrastructures and especially the peering fabrics of the large content providers in today's Internet. While [72, 84] provide only a qualitative description of Google's and Facebook's peering fabrics, [81] reports on a detailed analysis of proprietary data to identify the full set of peerings leveraged by Akamai to serve content to its end users. Our work is also concerned with identifying all peerings between a provider and the rest of the Internet, but in our case, the provider of interest is the largest cloud service provider (i.e., Amazon) and not a large CDN (i.e., Akamai), and instead of relying on proprietary data, our study only utilizes publicly available information.

In terms of methodology, our effort is similar to recent work described in [2, 55, 57, 62] which aims at developing techniques and tools for inferring inter-AS connections by solely relying on data-plane measurements in the form of traceroutes. Among the resulting tools, *MAP-IT* and *bdrmapIT* were developed as generic topology discovery tools, but as stated by the authors of [2, 57], these tools are not applicable within settings where layer-2 switching fabrics (not counting IXPs) are employed at the network borders. Since this assumption does not hold for cloud exchanges where today's VPIs are established, the tools cannot be used for our purpose. A third tool described in [55] is called *bdrmap* and appears to be directly applicable to our setting as it attempts to identify all inter-AS connections between a single network and the rest of the Internet. However, upon closer examination, we find that *bdrmap* is prone to produce inconsistent inference results (see § 8). Finally, `mi`^{2} is a new technique for inferring all interconnections at a given colocation facility and geo-locating them to the inside (or outside) of that facility [62]. However, because of the technique's inability to deal with layer-2 fabrics like cloud exchanges, the tool is not suitable for inferring cloud-specific interconnections. In other closely related work (e.g., see Chiu et al. [22]), cloud-centric probes were used to aid the discovery of AS paths and their length. In contrast, in this paper, we rely on cloud-centric probes to discover the peering fabric of the largest cloud provider and classify the identified peerings by their type, paying special attention to VPIs.

A recent study [41] casts pinning as a *constrained facility search (CFS)* problem and leverages various data sources (including targeted traceroute probes) to create sufficient constraints to pin an inferred interconnection to a single facility. Constrained-based search is certainly feasible for narrowing down possible colos for pinning Amazon's interconnections. However, the limited visibility of Amazon's peering in BGP (as we show in § 7) makes further probing by CFS-like approaches problematic. Furthermore, no code or implementation of CFS that is applicable to the pinning problem considered in this paper is available.

3 DATA COLLECTION & PROCESSING

To infer all peerings between Amazon and the rest of the Internet, we perform traceroute campaigns from Amazon's 15 available global regions to a .1 in each /24 prefix of the IPv4 address space.[3] To this end, we create a *t2-micro* instance VM within each of the 15 regions and break down the IPv4 address space into /24 prefixes. While we exclude broadcast and multicast prefixes, we deliberately consider addresses that are associated with private and shared address spaces since these addresses can be used internally in Amazon's own network. This process resulted in 15.6M target IPv4 addresses.

To probe these target IPs from our VMs, we use the Scamper tool [53] with UDP probes as they provide the highest visibility (i.e., response rate). Individual probes are terminated upon encountering five consecutive unresponsive hops in order to limit the overall measurement time while reaching Amazon's border routers. We empirically set our probing rate to 300pps to prevent blacklisting or rate control of our probe packets by Amazon. With this probing rate, our traceroute campaign took nearly 16 days to complete (from 08/03/2018 to 08/19/2018). Each collected traceroute is associated with a status flag indicating how the probe was terminated. We observed that the total number of completed traceroutes across different regions is fairly consistent but rather small (mean 7.7% and std $5 * 10^{-4}$) which suggests a limited yield. However, since our main objective is to identify Amazon interconnections and *not* to maximize traceroute yield, we consider any traceroute that leaves Amazon's network (i.e., reaches an IP outside of Amazon's network) as a candidate for revealing the presence of an interconnection, and the percentage of these traceroutes is about 77%.

Annotating Traceroute Data. To identify any Amazon interconnection traversed by our traceroutes, we annotate every IP hop with the following information: *(i)* its corresponding ASN, *(ii)* its organization (ORG), and *(iii)* whether it belongs to an IXP prefix. To map each IP address to its ASN, we rely on BGP snapshots from RouteViews and RIPE RIS (taken at the same time as our traceroute campaign). For ORG, we rely on CAIDA's AS-to-ORG dataset [47] and map the inferred ASN of each hop from the previous step to its unique ORG identifier. ORG information allows us to correctly identify the border interface of a customer in cases where traceroute traverses through hops in multiple Amazon ASes prior to reaching a customer network[4]. Finally, to determine if an IP hop is part of an IX prefix, we rely on PeeringDB [67], Packet Clearing House (PCH) [64], and CAIDA's IXP dataset [16] to obtain prefixes assigned to IXPs.

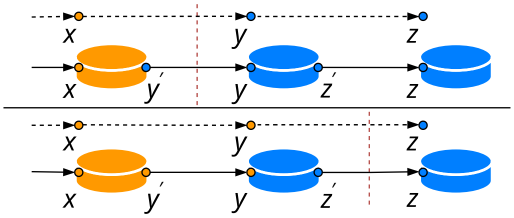


Figure 2: Illustrating potential error in detecting the interconnection segment between Amazon and a client when Amazon provides the IP addresses for both ends of the physical interconnection.

In our traceroutes, we observe IP hops that do not map to any ASN. These IPs can be divided into two groups. The first group consists of the IPs that belong to either a private or a shared address space (20.3%); we set the ASN of these IPs to 0. The second group consists of all the IPs that belong to the public address space but were not announced by any AS during our traceroute campaign (7%); for these IPs, we infer the AS owner by relying on WHOIS-provided information (i.e., name or ASN of the entity/company assigned by an RIR).

4 INFERRING INTERCONNECTIONS

In this section, we describe our basic inference strategy for identifying an Amazon-related interconnection segment across a given traceroute probe (§ 4.1) and discuss the potential ambiguity in the output of this strategy. We then discuss the extra steps we take to leverage these identified segments in an effort to efficiently expand the number of discovered Amazon-related interconnections (§ 4.2).

4.1 Basic Inference Strategy

Given the ASN-annotated traceroute data, we start from the source and sequentially examine each hop until we detect a hop that belongs to an organization *other* than Amazon (i.e., its ORG number is neither 0 nor 7224, which is Amazon). We refer to this hop as *customer border hop* and to its IP as a *Customer Border Interface (CBI)*. The presence of a *CBI* indicates that the traceroute has exited Amazon's network; that is, the traceroute hop right before a *CBI* is the *Amazon Border Interface (ABI)*, and the corresponding traceroute probe thus must have traversed an Amazon-related *interconnection segment*. For the remainder of our analysis, we only consider these initial portions of traceroutes between a source and an encountered *CBI*.[5] Next, for each *CBI*, we check to confirm that the AS owners of all the downstream hops in each traceroute does not include any ASN owned by Amazon (i.e., a sanity check that the traceroute does not re-enter Amazon); all of our traceroutes meet this condition. Finally, because of their unreliable nature, we exclude all traceroutes that contain either an (IP-level) loop, unresponsive hop(s) prior to Amazon's border, a *CBI* as the destination of a traceroute [11], or duplicate hops before Amazon's border. The first two rows of Table 1 summarize the number of *ABIs* and *CBIs* that we identified in our

[3] We observed a negligible difference in the visibility of interconnections across probes from different *AZs* in each region. Therefore, we only consider a single *AZ* from each region.

[4] We observed AS7224, AS16509, AS19047, AS14618, AS38895, AS39111, AS8987, and AS9059 for Amazon.

[5] In fact, we only need the *CBI* and the prior two *ABIs*.

Table 1: Number of unique *ABIs* and *CBIs* along with their fraction with various meta data, prior (rows 2-3) and after (rows 4-5) /24 expansion probing.

	All	BGP%	Whois%	IXP%
ABI	3.68k	38.4%	61.6%	-
CBI	21.73k	54.74%	24.8%	20.46%
eABI	3.78k	38.85%	61.15%	-
eCBI	24.75k	79.82%	2.32%	17.86%

traceroute data, along with the fraction of interfaces in each group for which we have BGP, Whois, and IXP-association information.

Ambiguity of Interconnection Segments. In certain cases, our basic strategy may not identify the correct Amazon-related interconnection segment on a given traceroute. To illustrate, consider traceroute probes that reveal the linear topology of three routers depicted in Figure 2. Suppose the physical link $IP_{y'} - IP_y$ between the left two routers represents the interconnection link. The assigned IP addresses for the interfaces y and y' should be from the same (/30 or /31) prefix that is provided either by the client (top) or Amazon (bottom). This is known as address sharing.[6] The color of observed interfaces (and routers) in Figure 2 indicates the inferred AS owner by our basic strategy (§ 4.1) in these two cases. Given that our traceroutes are always launched from Amazon to a client's network, this figure clearly shows that when Amazon provides addresses for the physical interconnection, our strategy incorrectly identifies the next downstream segment as an interconnection [5].

In summary, the described method always reveals the presence of an Amazon-related interconnection segment in a traceroute. The actual Amazon-specific interconnection segment is either the one between the identified ABI and CBI or the immediately preceding segment. Because of this ambiguity in accurately inferring the Amazon-specific interconnection segments, we refer to them as candidate *interconnection segments. In § 5, we present techniques for a more precise determination of these inferred candidate interconnection segments.*

4.2 Second Round of Probing to Expand Coverage

We perform our traceroute probes from each Amazon's region in two rounds. First, as described in § 4.1, we target .1 in each /24 prefix of the IPv4 address space (§ 3) and identify the pool of candidate interconnection segments. However, it is unlikely that our traceroute probes in this first round traverse through all the Amazon interconnections. Therefore, to increase the number of discovered interconnections, in a second round, we launch traceroutes from each region towards all other IP addresses in the /24 prefixes that are associated with each *CBI* that we discovered in the first round. Our reasoning for this "expansion probing" is that the IPs in these prefixes have a better chance to be allocated to *CBIs* than the IPs in other prefixes. Similar to round one, we annotate the resulting traceroutes and identify their interconnection segments (and the corresponding *ABIs* and *CBIs*). The bottom two rows in Table 1 show the total number of identified *ABIs* and *CBIs* after processing the collected expansion probes. In particular, while the first column of Table 1 shows a significant increase in the number of discovered *CBIs* (from 21.73k to 24.99k) and even some increase in the number of peering ASNs (from 3.52k to 3.55k) as a result of the expansion probing, the number of *ABIs* remains relatively constant.

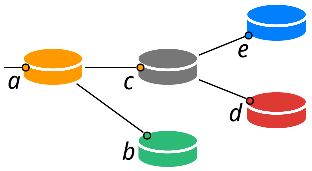


Figure 3: Illustration of a hybrid interface (a) that has both Amazon and client-owned interfaces as next hop.

5 VERIFYING INTERCONNECTIONS

To address the potential ambiguity in identifying the correct Amazon-specific segment of each inferred interconnection (§ 4.1), we first check these interconnections against three different heuristics (§ 5.1) and then rely on the router-level connectivity among border routers (§ 5.2) to verify (and possibly correct) the inferred *ABIs* and *CBIs*.

5.1 Checking Against Heuristics

We develop a few heuristics to check the aforementioned ambiguity of our approach with respect to inferring the correct interconnection segment. Since the actual interconnection segment could be the segment prior to the identified candidate segment (i.e., we might have to shift the interconnection to the previous segment), our heuristics basically check for specific pieces of evidence to decide whether an inferred *ABI* is correct or should be changed to its corresponding *CBI*. Once an *ABI* is confirmed, all of its corresponding *CBIs* are also confirmed. The heuristics are described below and are ordered (high to low) based on our level of confidence in their outcome.

IXP-Client. An IP address that is part of an IXP prefix always belongs to a specific IXP member. Therefore, if the IP address for a *CBI* in a candidate interconnection segment is part of an IXP prefix, then that *CBI* and its corresponding *ABI* are correctly identified [63].

Hybrid IPs. We observe *ABI* interfaces with hybrid connectivity. For example, in Figure 3, interface *a* represents such an interface with hybrid connectivity; it appears prior to the client interface *b* in one traceroute and prior to the Amazon interface *c* in another traceroute. Even if we are uncertain about the owner of an interface *c* (i.e., it may belong to the same or different Amazon client), we can reliably conclude that interface *a* has hybrid connectivity and must be an *ABI*.

Interface Reachability. Our empirical examination of traceroutes revealed that while *ABIs* are generally reachable from their corresponding clients, for security reasons, they are often not visible/reachable from the public Internet (e.g., a campus or residential networks). However, depending on the client configuration, *CBIs* may or may not be publicly reachable. Based on this empirical observation, we apply a heuristic that probes all candidate *ABIs* and *CBIs* from a vantage point in the public Internet (i.e., a node at the University of Oregon). Reachability (or unreachability) of a candidate *CBI* (or *ABIs*) from the public Internet offers independent evidence in support of our inference.

[6] This address sharing makes it even more difficult to accurately detect an interconnection segment between two ASNs in the middle of a traceroute [55].

Table 2: Number of candidate *ABIs* (and corresponding *CBIs*) that are confirmed by individual (first row) and cumulative (second row) heuristics.

	IXP	Hybrid	Reachable
Individual	0.83k (13.66k)	2.05k (14.44k)	2.8k (15.14k)
Cumulative	0.83k (13.66k)	2.26k (15.14k)	3.31k (24.23k)

Table 2 summarizes the fraction of identified *ABIs* (and thus their corresponding *CBIs*) that are confirmed by our individual (first row) and combined (second row) heuristics, respectively. We observe that our heuristics collectively confirmed 87.8% of all the inferred *ABIs* and thus 96.96% of the *CBIs*. The remaining 0.37k (or 9.81%) *ABIs* that do not match with any heuristic are interconnected with one (or multiple) *CBIs* that belong to a single organization. The resulting low rate of error in detecting the correct interconnection segments implies high confidence in the correctness of our inferred Amazon peerings.

5.2 Verifying Against Alias Sets

To further improve our ability to eliminate possible ambiguities in inferring the correct interconnection segments, we infer the router-level topology associated with all the candidate interconnections segments and determine the AS owner of individual routers. We consider any inferred interconnection segment to be correct if its *ABI* is on an Amazon router and its *CBI* is on a client router. In turn, for any incorrect segment, we first adjust the ownership of its corresponding *ABI* and *CBI* so as to be consistent with the determined router ownership and then identify the correct interconnection segment.

To this end, we utilize MIDAR [12] to perform alias resolution from VMs in all the regions where all the candidate *ABIs* and *CBIs* were observed. Each instance of this alias resolution effort outputs a set of (two or more) interfaces that reside on a single router. Given the potentially limited visibility of routers from different regions, we combine the alias sets from different regions that have any overlapping interfaces. Overall, we identify 2.64k alias sets containing 8.68k (2.31k *ABI* plus 6.37k *CBI*) interfaces and their sizes have a skewed distribution.

The direction of our traceroute probes (from Amazon towards client networks) and the fact that each router typically responds with the incoming interface suggest that the observed interfaces of individual Amazon (or client) border routers in our traceroute (i.e., IPs in each alias set) should typically belong to the same AS. This implies that there should be a majority AS owner among interfaces in an alias set. To identify the AS owner of each router, we simply examine the AS owner of individual IPs in the corresponding alias set. The AS that owns a clear majority of interfaces in an alias set is considered as the owner of the corresponding router and all the interfaces in the alias set.[7] We observe that for more than 94% (92%) of all alias sets, there is a single AS that owns >50% (100%) of all of an alias set's interfaces. The remaining 6% of alias sets comprises 343 interfaces with a median set size of 2. We consider the majority AS owner of each alias set as the AS owner of (all interfaces for) that router. Using this information, we check all of the inferred *ABIs* and *CBIs* to ensure that they are on a router owned by Amazon and the corresponding client, respectively. Otherwise, we change their labels. This consistency check results in changing the status of only 45 interfaces (i.e., 18, 2, and 25 change from *ABI* → *CBI*, *CBI* → *ABI*, and *CBI* → *CBI*[8], respectively). *These changes ultimately result in 3.77k ABIs and 24.76k CBIs associated with 4.02k unique ASes.*

6 PINNING INTERFACES

In this section, we first explore techniques to pin (i.e., geo-locate) each end of the inferred Amazon peerings (i.e., all *ABIs* and *CBIs*) to a specific colo facility, metro area, or a region and then evaluate our pinning methodology.

6.1 Methodology for Pinning

Our method for pinning individual interfaces to specific locations involves two basic steps. In a first step, we identify a set of border interfaces with known locations that we call *anchors*. Then, in a second step, we establish two *co-presence* rules to iteratively infer the location of individual unpinned interfaces based on the location of co-located anchors or other already pinned interfaces. That is, in each iteration, we propagate the location of pinned interfaces to their co-located unpinned neighbors.

Identifying Anchors. For *ABIs* or *CBIs* to serve as anchors for pinning other interfaces, we leverage the following four sources of information and consider them as reliable indicators of interface-specific locations.

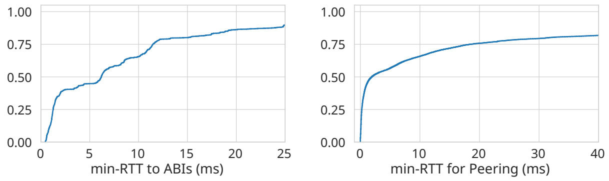
DNS Information (CBIs): A *CBI*[9] with specific location information embedded in its DNS name can be pinned to the corresponding colo or metro area. For example, a DNS name such as `ae-4.amazon.atlnga05.us.bb.gin.ntt.net` indicates that the *CBI* associated with NTT interconnects with Amazon in Atlanta, GA (*atlnga*). We use DNS parsing tools such as DRoP [46] along with a collection of hand-crafted rules to extract the location information (using 3-letter airport codes and full city names) from the DNS names of identified *CBIs*. In the absence of any ground truth, we check the inferred geolocation against the footprint of the corresponding AS from its PeeringDB listings or information on its webpage. Furthermore, we perform an RTT-constraint check using the measured RTTs from different Amazon regions to ensure that the inferred geolocation is feasible. This check, similar to DRoP [46], conservatively excludes 0.87k *CBIs* for which their inferred locations do not satisfy this RTT constraint.

IXP Association (CBIs): CBIs that are part of an IXP prefix can be pinned to the colo(s) in a metro area where the IXP is present. In total we have identified 671 IXPs within 471 (117) unique cities (countries) but exclude 10 IXPs (and their corresponding 366 *CBIs*) that are present in multiple metro areas as they cannot be pinned to a specific colo or metro area. Furthermore, we exclude all interfaces belonging to members that peer remotely. To determine those members, we first identified minIXRegion, the closest Amazon region to each IXP. We did this by measuring minIXRTT, the minimum

[7] We also examined router ownership at the organization level by considering all ASNs that belong to a single organization. This strategy allows us to group all Amazon/client interfaces regardless of their ASN to accurately detect the AS owner. However, since we observed one ASN per ORG in 99% of the identified alias sets, we present here only the owner AS of each router.

[8] This simply implies that the *CBI* interface belongs to another client.

[9] None of the *ABIs* had a reverse domain name associated with them.



(a)

(b)

Figure 4: (a) Distribution of min-RTT for *ABIs* from the closest Amazon region, and (b) Distribution of min-RTT difference between *ABI* and *CBI* for individual peering links.

RTT between the various regions and all interfaces that are part of the IXP and selecting minIXRegion as the Amazon region where minIXRTT is attained. Then we measure the minimum RTT between all interfaces and minIXRegion and label an interface as "local" if its RTT value is no more than 2ms higher than minIXRTT. We note that for about 80% of IXPs, the measured minIXRTT is less than 1.5ms (i.e., most IXPs are in very close proximity to at least one AWS region). This effort results in labeling about 2k out of the encountered 3.5k IXP interfaces in our measurements as "local." Conversely, there are some 1.5k interfaces belonging to members that peer remotely.

Single Colo/Metro Footprint (CBIs): CBIs of an AS that are present only at a single colo or at multiple colos in a given metro area can be pinned to that metro area. To identify those ASes that are only present in a single colo or a single metro area, we collect the list of all tenant ASes for 2.6k colo facilities from PeeringDB [52] as well as the list of all IXP participants from PeeringDB and PCH.

Native Amazon Colos (ABIs): Intuitively, *ABIs* that are located at colo facilities where Amazon is native (i.e., facilities that house Amazon's main border routers) must exhibit the shortest RTT from the VM in the corresponding region. To examine this intuition, we use two data sources for RTT measurements: *(i)* RTT values obtained through active probing [10] of *CBIs* and *ABIs*; and *(ii)* RTT values collected as part of the traceroute campaign. Figure 4a shows the distribution of the minimum RTT between VMs in different regions of Amazon and individual *ABIs*. We observe a clear knee at 2ms where around 40% of all the *ABIs* exhibit shorter RTT from a single VM. Given that all Amazon peerings have to be established through colo facilities where Amazon is native, we pin all these *ABIs* to the native colo closest to the corresponding VM. In some metro areas where Amazon has more than one native colo, we conservatively pinned the *ABIs* to the corresponding metro area rather than to a specific native colo.

Consistency Checking of Anchors. We perform two sets of consistency checks on the identified anchors. First, we check whether the inferred locations are consistent for those interfaces (i.e., 1.1k in total) that satisfy more than one of the four indicators we used to classify them as anchors. Second, we check for consistency across the inferred geolocation of different interfaces in any given alias set. These checks flagged a total of 66 (48 and 18) interfaces that had inconsistent geolocations and that we therefore excluded from our anchor list. These checks also highlight the conservative nature of our approach. In particular, by removing any anchors with inconsistent locations, we avoid the propagation of unreliable location information in our subsequent iterative pinning procedure (see below). The middle part of Table 3 presents the exclusive and cumulative numbers of *CBI* and *ABI* anchors (excluding the flagged ones) that resulted from leveraging the four utilized source of information.

Table 3: The exclusive and cumulative number of anchor interfaces by each type of evidence and pinned interfaces by our co-presence rules.

	Anchor Interface				Pinned Interface	
	DNS	**IXP**	**Metro**	**Native**	**Alias**	**min-RTT**
Exc.	5.31k	2.0k	1.66k	1.42k	0.65k	5.38k
Cum.	5.31k	6.73k	7.22k	8.64k	9.21k	14.37k

Inferring Co-located Interfaces. We use two co-presence rules to infer whether two interfaces are co-located in the same facility or same metro area. *(i) Rule 1 (Alias sets)*: This rule states that all interfaces in an alias set must be co-located in the same facility. Therefore, if an alias set contains one (or more) anchor(s), all interfaces in that set can be pinned to the location of that (those) anchor(s). *(ii) Rule 2 (Interconnections in a Single Metro Area)*: An Amazon peering is established between an Amazon border router and a client border router, and these routers are either in the same or in different colo/metro areas. Therefore, a small RTT between the two ends of an interconnection segment is an indication of their co-presence in at least the same metro area. The key issue is to determine a proper threshold for RTT delay to identify these co-located pairs. To this end, Figure 4b shows the distribution of the min-RTT differences between the two ends of all the inferred Amazon interconnection segments. While the min-RTT difference varies widely across all interconnection segments, the distribution exhibits a pronounced knee at 2ms, with approximately half of the inferred interconnection segments having min-RTT values less than this threshold. We use this threshold to separate interconnection segments that reside within a metro area (i.e., both ends are in the metro area) from those that extend beyond the metro area. Therefore, if one end of such a "short" interconnection segment is pinned, its other end can be pinned to the same metro area.

Iterative Pinning. Given a set of initial anchors at known locations as input, we identify and pin the following two groups of interfaces in an iterative fashion: *(i)* all unpinned alias sets that contain one (or more) anchor(s), and *(ii)* the unpinned end of all the short interconnection segments that have only one end pinned. For both steps, we extend our pinning knowledge to other interfaces *only if all anchors unanimously agree with the geolocation of the unpinned interface* [11]. This iterative process ends when there is no more interface that meets our co-presence rules. Our pinning process requires only four rounds to complete. The right-hand side of Table 3 summarizes the exclusive and cumulative number of interfaces pinned by each co-presence rule. Including all the anchors, *we are able to pin 45.05% (75.87%) of all the inferred CBIs (ABIs), and 50.21% of all border interfaces associated with Amazon's peerings.*

[10] This probing was done for a full day and used exclusively ICMP echo reply messages that can only be generated by intermediate hops and not by the target itself.

[11] We observed such a conflict in the propagation of pinning information only for 179 (1.2%) interfaces

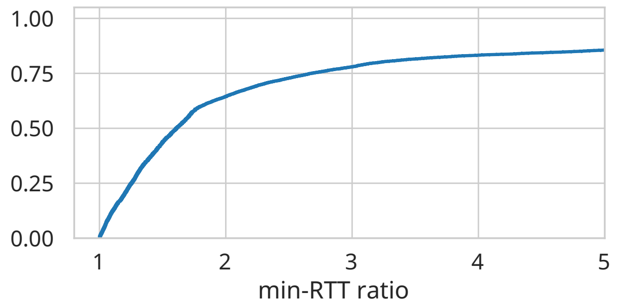


Figure 5: Distribution of the ratio of two lowest min-RTT from different Amazon regions to individual unpinned border interfaces.

Pinning at a Coarser Resolution. To better understand the reasons for being able to map only about half of all inferred interfaces associated with Amazon at the metro level, we next explore whether the remaining (14.21k) unpinned interfaces can be associated with a specific Amazon region based on their relative RTT distance. To this end, we examine the ratio of the two smallest min-RTT values for individual unpinned interfaces from each of the 15 Amazon regions. 1.11k of these interfaces are only visible from a single region and therefore the aforementioned ratio is not defined for these interfaces. We associate these interfaces to the only region from which they are visible. Figure 5 depicts the CDF of the ratio for the remaining (13.1k) unpinned interfaces that are reachable from at least two regions and shows that for 57% of these interfaces the ratio of two lowest min-RTT is larger than 1.5, i.e., the interface's RTT is 50% larger for one region. We map these interfaces to the region with the lowest delay. The relatively balanced min-RTT values for the remaining 43% of interfaces is mainly caused by the limited geographic separation of some regions. For example, the relatively short distance between *Virginia* and *Canada*, or between neighboring European countries makes it difficult to reliably associate some of the interfaces that are located between them using min-RTT values. *This coarser pinning strategy can map 8.67k (30.37%) of the remaining interfaces (0.62k ABIs and 8.05k CBIs) to a specific region which improves the overall coverage of the pinning process to a total of 80.58%.* However, because of the coarser nature of pinning, we do not consider these 30.37% of interfaces for the rest of our analysis and only focus on those 50.21% that we pinned at the metro (or finer) level.

6.2 Evaluation of Pinning

Accuracy. Given the lack of ground truth information for the exact location of Amazon's peering interfaces, we perform cross-validation on the set of identified anchors to enhance the confidence in our pinning results. Specifically, we perform a 10-fold stratified cross-validation with a 70-30 split for train-test samples. We employ stratified sampling [28] to maintain the distribution of anchors within each metro area and avoid cases where test samples are selected from metro areas with fewer anchors. We run our pinning process over the training set and measure both the number of pinned interfaces that match the test set (recall) and the number of pinned interfaces which agree geolocation-wise with the test set (precision). The results across all rounds are very consistent, with a mean value of 99.34% (57.21%) for precision (recall) and a standard deviation of $1.6*10^{-3}$ ($5.5*10^{-3}$). The relatively low recall can be attributed to the lack of known anchors in certain metro areas that prevented pinning information from propagating. The high precision attests to the conservative nature of our propagation technique (i.e., inconsistent anchors are removed and interfaces are only pinned when reliable (location) information is available) and highlights the low false positive rate of our pinning approach.

Geographic Coverage. We examine the coverage of our pinning results by comparing the cities where Amazon is known to be present against the metros where we have pinned border interfaces. Combining the reported list of served cities by Amazon [4] and the list of PeeringDB-provided cities [67] where Amazon establishes public or private peerings shows that Amazon is present in 74 metro areas. Our pinning strategy has geo-located Amazon-related border interfaces to 305 different metro areas across the world that cover all but three metro areas from Amazon's list, namely Bangalore (India), Zhongwei (China), and Cape Town (South Africa). While it is possible for some of our discovered, but unpinned *CBIs* to be located in these metros, we lack anchors in these three metros to reliably pin any interface to these locations. Finally, that our pinning strategy results in a significantly larger number of observed metros than the 74 metro areas reported by Amazon should not come as a surprise in view of the many inferred remote peerings where we have sufficient evidence to reliably pin the corresponding *CBIs*.

7 AMAZON'S PEERING FABRIC

In this section, we first present a method to detect whether an inferred Amazon-related interconnection is virtual (§ 7.1). Then we utilize various attributes of Amazon's inferred peerings to group them based on their type (§ 7.2) and reason about the differences in peerings across the identified groups (§ 7.3). Finally, we characterize the entire inferred Amazon connectivity graph (§ 7.4).

7.1 Detecting Virtual Interconnections

To identify private peerings that rely on virtual interconnections, we recall that a VPI is associated with a single (*CBI*) port that is utilized by a client to exchange traffic with one or more cloud providers (or other networks) over a layer-2 switching fabric. Therefore, a *CBI* that is common to two or more cloud providers must be associated with a VPI. Motivated by this observation, our method for detecting VPIs consists of the following three steps. First, we create a pool of target IP addresses that is composed of all identified non-IXP *CBIs* for Amazon, each of their *+1* next IP address, and all the destination IPs of those traceroutes that led to the discovery of individual unique *CBIs*. Second, we probe each of these target IPs from a number of major cloud providers other than Amazon and infer all the *ABIs* and *CBIs* along with the probes that were launched from these other cloud providers (using the methodology described in § 4). Finally, we identify any overlapping *CBIs* that were visible from two (or more) cloud providers and consider the corresponding interconnection to be a VPI. Note that this method yields a lower bound for the number of Amazon-related VPIs as it can only identify VPIs whose *CBIs* are visible from the considered cloud service providers. Any VPI that

Table 4: Number (and percentage) of Amazon's VPIs. These are *CBIs* that are also observed by probes originated from Microsoft, Google, IBM, and Oracle's cloud networks.

	Microsoft (%)	Google (%)	IBM (%)	Oracle (%)
Pairwise	4.69k (18.93)	0.79k (3.17)	0.23k (0.94)	0 (0)
Cumulative	4.69k (18.93)	4.93k (19.91)	5.01k (20.23)	5.01k (20.23)

is not used for exchanging traffic with multiple cloud provider is not identified by this method. Furthermore, we are only capable of identifying VPIs which utilize public IP addresses for their *CBIs* [5]. VPIs utilizing private addresses are confined to the virtual private cloud (VPC) of the customer and are not visible from anywhere within or outside of Amazon's network.

Applying this method, we probed nearly 327k IPs in our pool of target IP addresses from VMs in all regions of each one of the following four large cloud providers: Microsoft, Google, IBM, and Oracle. The results are shown in Table 4 where the first row shows the number of pairwise common *CBIs* between Amazon and other cloud providers. The second row shows the cumulative number of overlapping *CBIs*. From this table, we observe that roughly 20% of Amazon's *CBIs* are related to VPIs as they are visible from at least one other of the four considered cloud provider. While roughly 19% of VPIs are common between Amazon and Microsoft, there is no overlap in VPIs between Amazon and Oracle. Only 0.1% of Amazon's *CBIs* are common with Microsoft, Google and IBM.

Note that our method incorrectly identifies a VPI if a customer's border router is directly connected to Amazon but responds to our probe with a default or 3rd party interface. However, either of these two scenarios is very unlikely. For one, recall (§ 4) that we use UDP probes and do not consider a target interface as a *CBI* to avoid a response by the default interface [11]. Furthermore, our method selects *+1* IP addresses as traceroute targets (i.e., during the expansion probing) to increase the likelihood that the corresponding traceroutes cross the same *CBI* without directly probing the *CBI* itself. Also, the presence of a customer border router that responds with a third party interface implies that the customer relies on the third party for reaching Amazon while directly receiving downstream traffic from Amazon. However, such a setting is very unlikely for Amazon customers.

7.2 Grouping Amazon's Peerings

To study Amazon's inferred peering fabric, we first group all the inferred peerings/interconnections based on the following three key attributes: *(i)* whether the type of peering relationship is public or private, *(ii)* whether the corresponding AS link is present in public BGP feeds, and *(iii)* in the case of private peerings, whether the corresponding interconnection is physical or virtual (VPI). A peering is considered to be public (bi-lateral or multi-lateral) if its *CBI* belongs to an IXP prefix. We also check whether the corresponding AS relationship is present in the public BGP data by utilizing CAIDA's AS Relationships dataset [15] corresponding to the dates of our data collection. Although this dataset is widely used for AS relationship information, its coverage is known to be limited by the number and placement of BGP feed collectors (e.g., see [56] and references therein).

Table 5 gives the breakdown of all of Amazon's inferred peerings into six groups based on the aforementioned three attributes.

Table 5: Breakdown of all Amazon peerings based on their key attributes.

Group	ASes(%)	CBIs(%)	ABIs(%)
Pb-nB	2.52k (71)	3.93k (16)	0.79k (21)
Pb-B	0.20k (5)	0.56k (2)	0.56k (15)
Pb	*2.69k (76)*	*4.46k (18)*	*0.83k (22)*
Pr-nB-V	0.24k (7)	2.99k (12)	0.54k (14)
Pr-nB-nV	1.1k (31)	10.24k (41)	2.59k (69)
Pr-nB	*1.18k (33)*	*13.24k (53)*	*2.68k (71)*
Pr-B-nV	0.11k (3)	5.67k (23)	2.07k (55)
Pr-B-V	0.06k (2)	2.09k (8)	0.33k (9)
Pr-B	*0.12k (3)*	*7.76k (31)*	*2.11k (56)*

We use the labels Pr/Pb to denote private/public peerings, B/nB for being visible/not visible in public BGP feeds, and V/nV for virtual/non-virtual peerings (applies only in the case of private interconnections). For example, Pr-nB-nV refers to the number of Amazon's (unique) inferred private peerings that are not seen in public BGP feeds and are not virtual (e.g., cross connections). Each row in Table 5 shows the number (and percentage) of unique AS peers that establish certain types of peerings, along with the number (and percentage) of corresponding *CBIs* and *ABIs* for those peers. Since there are overlapping ASes and interfaces between different groups, Table 5 also presents three rows (i.e., rows 3, 6, and 9 with italic fonts) that aggregate the information for the two closely related prior pair of rows/groups. These three aggregate rows provide an overall view of Amazon's inferred peering fabric that highlight two points of general interest: *(i)* While 76% of Amazon's peers use Pb peering, only 33% of Amazon's peers use Pr-nB (virtual or physical) peerings, with the overlap of about 10% of peer ASes relying on both Pr-nB and Pb peerings, and the fraction of Pr-B peerings being very small (3%). *(ii)* The average number of *CBIs* (and *ABIs*) for ASes that use Pr-B, Pr-nB and Pb peerings to interconnect with Amazon is 65 (17), 11 (2), and 2 (0.3), respectively.

Hidden Peerings. Note that there are groups of Amazon's inferred peerings shown in Table 5 (together with their associated traffic) that remain in general hidden from the measurement techniques that are commonly used for inferring peerings (e.g., traceroute). One such group consists of all the virtual peerings (Pr-*-V) since they are used to exchange traffic between customer ASes of Amazon (or their downstream ASes) and Amazon. The second group is made up of all other non-virtual peerings that are not visible in BGP data, namely Pr-nB-nV and even Pb-nB. The presence of these peerings cannot be inferred from public BGP data and their associated traffic is only visible along the short AS path to the customer AS. These hidden peerings make up 33.29% of all of Amazon's inferred peerings and their associated traffic is carried over Amazon's private backbone and not over the public Internet.

Table 6: Hybrid peering groups along with the number of unique ASes for each group.

Different Types of Hybrid Peering	#ASN
Pb-nB	2183
Pr-nB-nV	730
Pr-nB-nV; Pb-nB	207
Pb-B	117
Pr-nB-nV; Pr-nB-V	87
Pr-nB-nV; Pb-nB; Pr-nB-V	60
Pr-nB-V	45
Pb-nB; Pr-nB-V	41
Pr-B-nV	30
Pb-B; Pb-nB	27
Pb-B; Pr-B-nV	25
Pb-B; Pr-B-V; Pr-B-nV	24
Pr-B-nV; Pr-B-V	18
Pr-nB-nV; Pr-B-V; Pr-B-nV	5
Pb-B; Pr-B-V	4
Pr-B-V	4
Pb-B; Pr-B-V; Pr-B-nV; Pr-nB-nV	2
Pr-nB-nV; Pr-B-nV; Pr-nB-V	1
Pr-nB-nV; Pr-B-nV	1
Pb-B; Pr-B-nV; Pr-nB-nV	1
Pb-B; Pr-B-nV; Pr-nB-V; Pr-B-V; Pr-nB-nV	1

Hybrid Peering. Individual ASes may establish multiple peerings of different types (referred to as "hybrid" peering) with Amazon; that is, appear as a member of two (or more) groups in Table 5. We group all ASes that establish such hybrid peering based on the combination of peering types that are listed in Table 5 types and that they maintain with Amazon. The following are two of the most common hybrid peering scenarios we observe. **Pr-nB-nV + Pb-nB:** With 207 ASes, this is the largest group of ASes which utilize hybrid peering. Members of this group use both types of peerings to exchange their own traffic with Amazon and include ASes such as Akamai, Intercloud, Datapipe, Cloudnet, and Dell. **Pr-nB-nV; Pb-nB; Pr-nB-V:** This group is similar to the first group one but its members also utilize virtual peerings to exchange their own traffic with Amazon. This group consists of 60 ASes that include large providers such as Google, Microsoft, Facebook, and Limelight. Table 6 gives a detailed breakdown of the observed hybrid (and non-hybrid) peering groups and shows for each group the number of ASes that use that peering group. Note that each AS is counted only once in the group that has the most specific peering types.

7.3 Inferring the Purpose of Peerings

In an attempt to gain insight into how each of the six different groups of Amazon's peerings is being used in practice, we consider a number of additional characteristics of the peers in each group and depict those characteristics using stacked boxplots as shown in Figure 6. In particular, starting with the top row in Figure 6, we consider summary distributions of [12] *(i) size of customer cone of peering AS* (i.e., number of /24 prefixes that are reachable through the AS (labeled as "BGP /24"); *(ii)* number of /24 prefixes that are reachable from Amazon through the identified *CBIs* associated with each peering; *(iii)* number of *ABIs* for individual peering AS; *(iv)* number of *CBIs* for individual peering AS; *(v)* min RTT difference between both ends of individual peering; *(vi)* number of unique metro areas that the *CBIs* of each peering AS have been pinned to (see § 6).

For example, we view the number of /24 prefixes in the customer cone of an AS to reflect the AS's size/role (i.e., as tier-1 or tier-2 AS) in routing Internet traffic. Moreover, comparing the number of /24 prefixes in the customer cone with the number of reachable /24 prefixes through a specific peering for an AS reveals the purpose of the corresponding peering to route traffic to/from Amazon from/to its downstream networks. In the following, we discuss how the combined information in Table 5 and Figure 6 sheds light on Amazon's global-scale peering fabric and illuminates the different roles of the six groups of peering ASes.

Pb-nB. The peers in this group are typically edge networks with a small customer cone (including content, enterprise, and smaller transit/access networks) that exchange traffic with Amazon through a single *CBI* at an IXP. The corresponding routes are between Amazon and these edge networks and are thus *not* announced in BGP. Peers in this group include CDNs like Akamai, small transit/access providers like Etisalat, BT, and Floridanet, and enterprises such as Adobe, Cloudflare, Datapipe (Rackspace), Google, Symantec, LinkedIn, and Yandex.

Pb-B. This group consists mostly of tier-2 transit networks with moderate-sized customer cones. These networks are present at a number of IXPs to connect their their downstream customer networks to Amazon. The corresponding routes must be announced to downstream ASes and are thus visible in BGP. Example peers in this group are CW, DigitalOcean, Fastweb, Seabone, Shaw Cable, Google Fiber, and Vodafone.

Pr-nB-V. The peers in this group are a combination of small transit providers and some content and enterprise networks. They establish VPIs at a single location to exchange either their own traffic or the traffic of their downstream networks with Amazon through a VPI. Therefore, their peering is not visible in BGP. About 85% of these peers are visible from two cloud providers while the rest is visible from more than two cloud providers. Examples of enterprise and content networks in this group are Apple, UCSD, UIOWA, LG, and Edgecast, and examples of transit networks are Rogers, Charter, and CenturyLink.

Pr-nB-nV. These peers appear to establish physical interconnections (i.e., cross-connects) with Amazon since they are not reachable from other cloud providers. However, given the earlier-mentioned under-counting of VPIs by our method, we hypothesize that some or all of these peerings could be associated with VPIs, similar to the previous group. The composition of the peers in this group is comparable to **Pr-nB-V** but includes a larger fraction of enterprise networks (i.e., main users of VPIs) which in turn is consistent with our hypothesis. Examples of peers in this group are enterprises such as Datapipe (Rackspace), Chevron, Vox-Media, UToronto, and Georgia-Tech, CDNs such as Akamai and Limelight and transit/access providers like Comcast. To further examine our hypothesis, we parse the DNS names of 4.85k *CBIs* associated with peers in the **Pr-nB** group. 170 of these DNS names (100 from

[12] For ASes that utilize hybrid peering with Amazon, the reported information in each group only includes peerings related to that group.

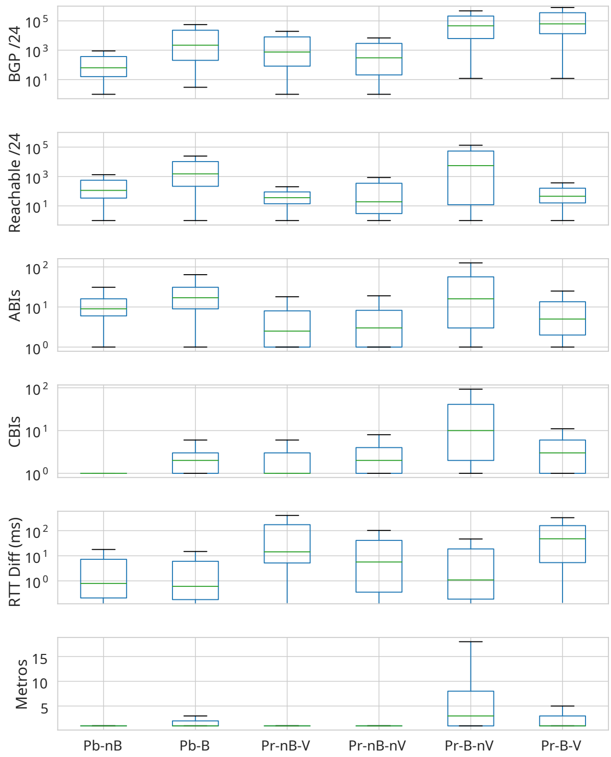


Figure 6: Key features of the six groups of Amazon's peerings (presented in Table 5) showing (from top to bottom): the number of /24 prefixes within the customer cone of peering AS, the number of probed /24 prefixes that are reachable through the *CBIs* of associated peerings of an AS, the number of *ABIs* and *CBIs* of associated of an AS, the difference in RTT of both ends of associated peerings of an AS, and the number of metro areas which the *CBIs* of each peering AS have been pinned to.

Pr-nB-nV and 70 from Pr-nB-V interfaces) contain VLAN tags, indicating the presence of a virtual private interconnection. We also observe some commonly used (albeit not required) keywords [7] such as *dxvif* (Amazon terminology for "direct connect virtual interface"), *dxcon, awsdx* and *aws-dx* for 125 (out of 170) *CBIs* where the "dx"-notation is synonymous with an interface's use for "direct interconnections". We consider the appearance of these keywords in the DNS names of *CBIs* for this group of peerings (and only in this group) as strong evidence that the interconnections in question are indeed VPIs. Therefore, a subset of Pr-nB-nV interconnections is likely to be virtual as well.

Pr-B-nV. The peers in this group are very large transit networks that establish cross-connections at various locations (many *CBIs* and *ABIs*) across the world). The large number of prefixes that are reachable through them from Amazon and the visibility of the peerings in BGP suggest that these peers simply provide connectivity for their downstream clients to Amazon. Given the large size of these transit networks, the visibility of these peerings in BGP is due to the announcement of routes from Amazon to all of their downstream networks. Intuitively, given the volume of aggregate traffic exchanged between Amazon and these large transit networks, the peers in this group have the largest number of *CBIs*, and these *CBIs* are located at different metro areas across the world. Example networks in this group are AT&T, Level3 (now CenturyLink), GTT, Cogent, HE, XO, Zayo, and NTT.

Pr-B-V. This group consists mostly a subset of the very large transit networks in **Pr-B-nV** and the peers in this group also establish a few VPIs (at different locations) with Amazon. The small number of prefixes that are reachable from Amazon through these peers along with the large number of *CBIs* per peer indicates that these peers bring specific Amazon clients (a provider or enterprise, perhaps even without an ASN) to a colo facility to exchange traffic with Amazon [6]. The presence of these peerings in BGP is due to the role they play as transit networks in the **Pr-B-nV** group that is separate from peers in this group using virtual peerings. Example networks in this group are Cogent, Comcast, CW, GTT, CenturyLink, HE, and TimeWarner, all of which are listed as Amazon cloud connectivity partners [6, 44, 59]) and connect enterprises to Amazon. When examining the min RTT difference between both ends of peerings across different groups (row 5 in Figure 6), we observe that both groups with virtual interconnections (Pr-B-V and Pr-nB-V) have in general larger values than the other groups. This observation is in agreement with the fact that many of these VPIs are associated with enterprises that are brought to the cloud exchange by access networks using layer-2 connections.

Coverage of Amazon's Interconnections. Although the total number of peerings that Amazon has with its customers is not known, our goal here is to provide a baseline comparison between Amazon's peering fabric that is visible in public BGP data and Amazon's peering fabric as inferred by our approach. Using our approach, we have identified 3.3k unique peerings for Amazon. In contrast, there are only 250 unique Amazon peerings reported in BGP, and 226 of them are also discovered by our approach. Upon closer examination, for some of the 24 peerings that are seen in BGP but not by our approach, we observed a sibling of the corresponding peer ASes. This brings the total coverage of our method to about 93% of all reported Amazon peerings in BGP. In addition, we report on more than 3k unique Amazon peerings that are not visible in public BGP data. These peerings with Amazon and their associated traffic are not visible when relying on more conventional measurement techniques.

7.4 Characterizing Amazon's Connectivity Graph

Having focused so far on groups of peerings of certain types or individual AS peers, we next provide a more holistic view of Amazon's inferred peering graph and examine some of its basic characteristics. We first produce the Interface Connectivity Graph (ICG) between all the inferred border interfaces. ICG is a bipartite graph where each node is a border interface (an *ABI* or a *CBI*) and each edge corresponds to the traceroute interconnection segment (ICS) between an *ABI* and a *CBI*. We also annotate each edge with the difference in the minimum RTT from the closest VM to each end of the ICS.[13] Intuitively, we expect the resulting ICG to have a separate partition

[13] We identify the VM that has the shortest RTT from an *ABI* and use the min-RTT of the same VM from the corresponding *CBI* to determine the RTT of an ICS.

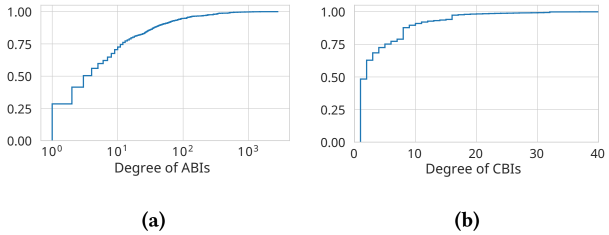


Figure 7: Distribution of *ABIs* (log scale) and *CBIs* degree in left and right figures accordingly.

that consists of interconnections associated with each region, i.e., *ABIs* of a region connecting to *CBIs* that are supported by them. However, we observe that the ICG's largest connected component consists of the vast majority (92.3%) of all nodes. This implies that there are links between *ABIs* in each Amazon region and *CBIs* in several other regions. Upon closer examination of 57.85% of all the peerings that have both of their ends pinned, we notice that a majority of these peerings (98%) are indeed contained within individual Amazon regions. However, we do encounter remote peerings between regions that are a significant geographical distance apart. For example, there are peerings between FR and KR, US-VA and SG, AU and CA. The large fraction of peerings with only one end or no end pinned (about 42%) suggests that the actual number of remote peerings is likely to be much larger. These remote peerings are the main reason for why the ICG's largest connected component contains more than 92% of all border interfaces.

To illustrate the basic connectivity features of the bi-partite ICG, Figures 7a and 7b show the distributions of the number of *CBIs* that are associated with each individual *ABIs* (degree of *ABIs*) and the number of *ABIs* associated with individual *CBIs* (degree of *CBIs*). We observe a skewed distribution for *ABI* degree where 30%, 70%, and 95% of *ABIs* are associated with 1, <10, and <100 *CBIs*, respectively. Roughly 50% (90%) of *CBIs* are associated with a single (≤ 8) *ABIs*. A closer examination shows that high degree *CBIs* are mainly associated with Amazon's public peerings with large transit networks (e.g., GTT, Cogent, NTT, CenturyLink). In contrast, a majority of high degree *ABIs* is associated with private, non-BGP, non-virtual peerings (see § 7).

8 INFERRING PEERING WITH BDRMAP

As stated earlier in § 2, *bdrmap* [55][14] is the only other existing tool for inferring border routers of a given network from traceroute data. With Amazon as the network of interest, our setting appears to be a perfect fit for the type of target settings assumed by *bdrmap*. However, there are two important differences between the cloud service provider networks we are interested in (e.g., Amazon) and the more traditional service provider network that *bdrmap* targets (e.g., a large US Tier-1 network). First, not only can the visibility of different prefixes vary widely across different Amazon regions, but roughly one-third of Amazon's peerings are not visible in BGP and even some of the BGP-visible peerings of a network are related to other instances of its peerings with Amazon (§ 7). At the same time, *bdrmap* relies on peering relationships in BGP to determine the targets for its traceroute probes and also uses them as input for some of its heuristics. Therefore, *bdrmap*'s outcome is affected by any inconsistent or missing peering relationship in BGP. Second, as noted earlier, our traceroute probes reveal hybrid Amazon border routers that have both Amazon and client routers as their next hop and connect to them. This setting is not consistent with *bdrmap*'s assumption that border routers should be situated exclusively in the host or peering network. Given these differences, the comparison below is intended as a guideline for how *bdrmap* could be improved to apply in a cloud-centric setting.

Thanks to special efforts by the authors of bdrmap who modified their tool so it could be used for launching traceroutes from cloud-based vantage points (i.e., VMs), we were able to run it in all Amazon regions to compare the bdrmap-inferred border routers with our inference results. *bdrmap* identified 4.83k *ABIs* and 9.65k *CBIs* associated with 2.66k ASes from all global regions. 3.23k of these *CBIs* belong to IXP prefixes and are associated with 1.81k ASes. Given *bdrmap*'s customized probing strategy and its extensive use of different heuristics, it is not feasible to identify the exact reasons for all the observed differences between *bdrmap*'s and our findings. However, we were able to identify the following three major inconsistencies in *bdrmap*'s output.

First, *bdrmap* does not report an AS owner for 0.32k of its inferred *CBIs* (i.e., owner is AS0). Second, instances of *bdrmap* that run in different Amazon regions report different AS owners for more than 500 *CBIs*, sometimes as many as 4 or 5 different AS owners for an interface. Third, running instances of *bdrmap* in different Amazon regions results in inconsistent views of individual border router interfaces; e.g., one and the same interface is inferred to be an *ABI* from one region and a *CBI* from another region. We identified 872 interfaces that exhibit this inconsistency. Furthermore, the fact that 97% (846 out of 872) of the interfaces with this type of inconsistency are advertised by Amazon's ASNs indicates that the AS owner for these interfaces have been inferred by *bdrmap*'s heuristics.

When comparing the findings of *bdrmap* against our methodology in more detail, we observed that our methodology and *bdrmap* have 1.85k, 5.48k, and 2k *ABI*, *CBI*, and ASes in common. However, without access to ground truth, a full investigation into the various points of disagreement is problematic. To make the problem more tractable, we limit our investigation to the 0.65k ASes that were exclusively identified by *bdrmap* and try to rely on other sources of information to confirm or dismiss *bdrmap*'s findings. These exclusive ASNs belong to 0.18k (0.49k) IXP (private) peerings. For IXP peerings, we compare *bdrmap*'s findings against IP-to-ASN mappings that are published by IXP operators or rely on embedded information within DNS names. The inferences of *bdrmap* is only aligned for 42 of these peers. For the 0.49k private peerings we focus on inferences that were made by the *thirdparty* heuristic as it constitutes the largest (62%) fraction of *bdrmap*-exclusive private peerings (for details, see § 5.4 in [55]). These ASes are associated with 375 *CBIs* and we observe 66 (60 ASNs) of these interfaces in our data. For each of these 66 *CBIs*, we calculate the set of reachable destination ASNs through these *CBIs* and determine the upstream provider network for each one of these destination ASes using BGP data [15]. Observing more than one or no common provider network among reachable destination ASes for individual *CBIs* would invalidate the application of *bdrmap*'s *thirdparty* heuristic, i.e., *bdrmap* wouldn't

[14] *MAP-IT* [57] and *bdrmapIT* [2] are not suitable for this setting since we have layer-2 devices at the border.

have applied this heuristic if it had done more extensive probing that revealed an additional set of reachable destination ASes for these *CBIs*. We find that 50 (44 ASNs) out of the 66 common *CBIs* have more than one or no common providers for the target ASNs. Note that this observation does not invalidate *bdrmap*'s *thirdparty* heuristics but highlights its reliance on high-quality BGP snapshots and AS-relationship information.

9 LIMITATIONS OF OUR STUDY

As a third-party measurement study of Amazon's peering fabric that makes no use of Amazon-proprietary data and only relies on generally-available measurement techniques, there are inherent limitations to our efforts aimed at inferring and geo-locating all interconnections between Amazon and the rest of the Internet. This section collects and organizes the key limitations in one place and details their impact on our findings.

Inferring Interconnections. Border routers responding to traceroute probes using a third-party address are a well-known cause for artifacts in traceroute measurement output, and our IXP-client and Hybrid-IP heuristics used in § 5.1 are not immune to this problem. However, as reported in [54], the fraction of routers that respond with their incoming interface is in general above 50% and typically even higher in the U.S.

In contrast, because of the isolation of network paths for VPIs of Amazon's clients that use private addresses, any peerings associated with these VPIs are not visible to probes from VMs owned by other Amazon customers. As a result, our inference methodology described in § 4 cannot discover established VPIs that leverage private IP addresses.

Pinning Interconnections. In § 6, we reported being able to pin only about half of all the inferred peering interfaces at the metro level. In an attempt to understand what is limiting our ability to pin the rest of the inferred interfaces, we identified two main reasons. First, there is a lack of anchors in certain regions, and second, there is the common use of remote peering. These two factors in conjunction with our conservative iterative strategy for pinning interfaces to the metro level make it difficult to provide enough and sufficiently reliable indicators of interface-specific locations.

One way to overcome some of these limiting factors is by using a coarser scale for pinning (e.g., regional level). In fact, as shown in § 6, at the regional level, we are able to pin some 30% of the remaining interfaces which improves the overall coverage of our pinning strategy at the granularity of regions to about 80%.

Other Observations. Although our study does not consider IPv6 addresses, we argue that the proposed methodology only requires minimal modifications (e.g., incorporating IPv6 target selection techniques [13, 38]) to be applicable to infer IPv6 peerings. We will explore IPv6 peerings as part of future work.

Like others before us, as third-party researchers, we found it challenging to validate our Amazon-specific findings. Like most of the large commercial provider networks, Amazon makes little, if any, ground truth data about its global-scale serving infrastructure publicly available, and our attempts at obtaining peering-related ground truth information from either Amazon, Amazon's customers, operators of colo facilities where Amazon is native, or AWS Direct Connect Partners have been futile.

Faced with the reality of a dearth of ground truth data, whenever possible, we relied on extensive consistency-checking of our results (e.g., see § 5, § 6). At the same time, many of our heuristics are conservative in nature, typically requiring agreement when provided with input from multiple complementary sources of information. As a result, the reported quantities in this paper are in general lower bounds but nevertheless demonstrate the existence of a substantial number of Amazon-related peerings that are not visible to more conventional measurement studies and/or inference techniques.

10 SUMMARY

In this paper, we present a measurement study of the interconnection fabric that Amazon utilizes in the US to run its various businesses, including AWS. We show that in addition to some 0.12k private peerings and about 2.69k pubic peerings (i.e., bi-lateral and multi-lateral peerings), Amazon also utilizes at least 0.24k (and likely many more) virtual private interconnections or VPIs. VPIs are a new and increasingly popular interconnection option for entities such as enterprises that desire highly elastic and flexible connections to the cloud providers that offer the type of services that these entities deem critical for running their business. Our study makes no use of Amazon-proprietary data and can be used to map the interconnection fabric of any large cloud provider, provided the provider in question does not filter traceroute probes.

Our findings emphasize that new methods are needed to track and study the type of "hybrid" connectivity that is in use today at the Internet's edge. This hybrid connectivity describes an emerging strategy whereby one part of an Internet player's traffic bypasses the public Internet (i.e., cloud service-related traffic traversing cloud exchange-provided VPIs), another part is handled by its upstream ISP (i.e., traversing colo-provided private interconnections), and yet another portion of its traffic is exchanged over a colo-owned and colo-operated IXP. As the number of businesses investing in cloud services is expected to continue to increase rapidly, multi-cloud strategies are predicted to become mainstream, and the majority of future workload-related traffic is anticipated to be handled by cloud-enabled colos [37], tracking and studying this hybrid connectivity will require significant research efforts on parts of the networking community. Knowing the structure of this hybrid connectivity, for instance, is a prerequisite for studying which types of interconnections will handle the bulk of tomorrow's Internet traffic, and how much of that traffic will bypass the public Internet, with implications on the role that traditional players such as Internet transit providers and emerging players such as cloud-centric data center providers may play in the future Internet.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and our shepherd, Cristel Pelsser, for their insightful feedback. We would also like to thank Matthew Luckie for help and feedback regarding the use of *bdrmap* [55] in cloud environments as well as Albert Greenberg and Karthick Jayaraman for helping us to conduct measurements from Microsoft's Azure platform. This material is based upon work supported by the National Science Foundation under Awards CNS-1320977 and CNS-1719165.

REFERENCES

[1] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. 2012. Anatomy of a large European IXP. *SIGCOMM CCR* (2012).

[2] Marder Alexander, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, Claffy KC, and Smith M. Jonathan. 2018. Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale. In *Internet Measurement Conference (IMC)*. ACM.

[3] Amazon. 2018. AWS Direct Connect. https://aws.amazon.com/directconnect/.

[4] Amazon. 2018. AWS Direct Connect | Product Details. https://aws.amazon.com/directconnect/details/.

[5] Amazon. 2018. AWS Direct Connect Frequently Asked Questions. https://aws.amazon.com/directconnect/faqs/.

[6] Amazon. 2018. AWS Direct Connect Partners. https://aws.amazon.com/directconnect/partners/.

[7] Amazon. 2018. Describe Virtual Interfaces. https://docs.aws.amazon.com/cli/latest/reference/directconnect/describe-virtual-interfaces.html.

[8] Amazon. 2018. Regions and Availability Zones - Amazon Elastic Compute Cloud. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-regions-availability-zones.

[9] David G Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan. 2002. Topology inference from BGP routing dynamics. In *Internet Measurement Conference (IMC)*. ACM.

[10] Brice Augustin, Balachander Krishnamurthy, and Walter Willinger. 2009. IXPs: Mapped?. In *Internet Measurement Conference (IMC)*. ACM.

[11] Fred Baker. 1995. *Requirements for IP Version 4 Routers*. Technical Report. Cisco Systems.

[12] Adam Bender, Rob Sherwood, and Neil Spring. 2008. Fixing ally's growing pains with velocity modeling. In *SIGCOMM*. ACM.

[13] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P Rohrer. 2018. In the IP of the beholder: Strategies for active IPv6 topology discovery. In *Internet Measurement Conference (IMC)*. ACM.

[14] Ingrid Burrington. 2016. Why Amazon's Data Centers Are Hidden in Spy Country. https://www.theatlantic.com/technology/archive/2016/01/amazon-web-services-data-center/423147/.

[15] CAIDA. 2018. AS Relationships. http://www.caida.org/data/as-relationships/.

[16] CAIDA. 2018. The CAIDA UCSD IXPs Dataset. http://www.caida.org/data/ixps.xml.

[17] Matt Calder, Xun Fan, Zi Hu, Ethan Katz-Bassett, John Heidemann, and Ramesh Govindan. 2013. Mapping the Expansion of Google's Serving Infrastructure. In *Internet Measurement Conference (IMC)*. ACM.

[18] Ignacio Castro, Juan Camilo Cardona, Sergey Gorinsky, and Pierre Francois. 2014. Remote Peering: More Peering without Internet Flattening. In *CoNEXT*. ACM.

[19] Joseph Chabarek and Paul Barford. 2013. What's in a name?: decoding router interface names. In *HotPlanet*. ACM.

[20] H Chang. 2006. *Modeling Internet's Inter-Domain Topology and Traffic Demand Based on Internet Business Characterization*. Ph.D. Dissertation. PhD thesis, University of Michigan.

[21] Nikolaos Chatzis, Georgios Smaragdakis, Jan Böttger, Thomas Krenc, and Anja Feldmann. 2013. On the Benefits of Using a Large IXP as an Internet Vantage Point. In *Internet Measurement Conference (IMC)*. ACM.

[22] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, and Ramesh Govindan. 2015. Are we one hop away from a better internet?. In *Internet Measurement Conference (IMC)*. ACM.

[23] CoreSite. 2018. The Coresite Open Cloud Exchange - One Connection. Countless Cloud Options. https://www.coresite.com/solutions/cloud-services/open-cloud-exchange.

[24] DatacenterMap. 2018. Amazon EC2. http://www.datacentermap.com/cloud/amazon-ec2.html.

[25] Yuri Demchenko, Jeroen Van Der Ham, Canh Ngo, Taras Matselyukh, Sonja Filiposka, Cees de Laat, and Eduard Escalona. 2013. Open cloud exchange (OCX): Architecture and functional components. In *Cloud Computing Technology and Science*. IEEE.

[26] Amogh Dhamdhere and Constantine Dovrolis. 2010. The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *CoNEXT*. ACM.

[27] Dhamdhere, Amogh and Dovrolis, Constantine. 2011. Twelve years in the evolution of the Internet ecosystem. *Transactions on Networking (ToN)* (2011).

[28] NA Diamantidis, Dimitris Karlis, and Emmanouel A Giakoumakis. 2000. Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence* (2000).

[29] Xenofontas A Dimitropoulos, Dmitri V Krioukov, and George F Riley. 2005. Revisiting Internet AS-level topology discovery. In *Passive and Active Measurement (PAM)*. Springer.

[30] Benoit Donnet and Timur Friedman. 2007. Internet topology discovery: a survey. *Communications Surveys & Tutorials, IEEE* (2007).

[31] Ramakrishnan Durairajan, Subhadip Ghosh, Xin Tang, Paul Barford, and Brian Eriksson. 2013. Internet atlas: a geographic database of the internet. In *HotPlanet*. ACM.

[32] Ramakrishnan Durairajan, Joel Sommers, and Paul Barford. 2014. Layer 1-Informed Internet Topology Measurement. In *Internet Measurement Conference (IMC)*. ACM.

[33] Ramakrishnan Durairajan, Joel Sommers, Walter Willinger, and Paul Barford. 2015. InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure. In *SIGCOMM*. ACM.

[34] Equinix. 2017. Cloud Exchange. http://www.equinix.com/services/interconnection-connectivity/cloud-exchange/.

[35] B. Eriksson, P. Barford, B. Maggs, and R. Nowak. 2012. Posit: A Lightweight Approach for IP Geolocation. *SIGMETRICS Performance Evaluation Review* (2012).

[36] D. Feldman, Y. Shavitt, and N. Zilberman. 2012. A structural approach for PoP GEO-location. *Computer Networks* (2012).

[37] Gartner. 2016. https://www.gartner.com/doc/3396633/market-trends-cloud-adoption-trends.

[38] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the expanse: Understanding and unbiasing IPv6 hitlists. In *Internet Measurement Conference (IMC)*. ACM.

[39] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. 2008. The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse?. In *Passive and Active Measurement (PAM)*. Springer.

[40] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, et al. 2014. Inferring Complex AS Relationships. In *Internet Measurement Conference (IMC)*. ACM.

[41] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and k. claffy. 2015. Mapping Peering Interconnections to a Facility. In *CoNEXT*. ACM.

[42] Vasileios Giotsas, Shi Zhou, Matthew Luckie, et al. 2013. Inferring Multilateral Peering. In *CoNEXT*. ACM.

[43] Google. 2018. GCP Direct Peering. https://cloud.google.com/interconnect/docs/how-to/direct-peering.

[44] Google. 2018. Partner Interconnect | Google Cloud. https://cloud.google.com/interconnect/partners/.

[45] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. 2006. Constraint-Based Geolocation of Internet Hosts. *Transactions on Networking (ToN)* (2006).

[46] B. Huffaker, M. Fomenkov, and k. claffy. 2014. DRoP:DNS-based Router Positioning. *SIGCOMM CCR* (2014).

[47] Bradley Huffaker, Ken Keys, Marina Fomenkov, and Kimberly Claffy. 2018. AS-to-Organization Dataset. http://www.caida.org/research/topology/as2org/.

[48] E. Katz-Bassett, J. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. 2006. Towards IP Geolocation Using Delay and Topology Measurements. In *Internet Measurement Conference (IMC)*. ACM.

[49] Ken Keys. 2010. Internet-Scale IP Alias Resolution Techniques. *SIGCOMM CCR* (2010).

[50] Ken Keys, Young Hyun, Matthew Luckie, and Kim Claffy. 2013. Internet-Scale IPv4 Alias Resolution with MIDAR. *Transactions on Networking (ToN)* (2013).

[51] Akmal Khan, Taekyoung Kwon, Hyun-chul Kim, and Yanghee Choi. 2013. AS-level topology collection through looking glass servers. In *Internet Measurement Conference (IMC)*. ACM.

[52] Aemen Lodhi, Natalie Larson, Amogh Dhamdhere, Constantine Dovrolis, et al. 2014. Using peeringDB to understand the peering ecosystem. *SIGCOMM CCR* (2014).

[53] Matthew Luckie. 2010. Scamper: a scalable and extensible packet prober for active measurement of the internet. In *Internet Measurement Conference (IMC)*. ACM.

[54] Matthew Luckie et al. 2014. A second look at detecting third-party addresses in traceroute traces with the IP timestamp option. In *Passive and Active Measurement (PAM)*. Springer.

[55] Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, David Clark, et al. 2016. bdrmap: Inference of Borders Between IP Networks. In *Internet Measurement Conference (IMC)*. ACM.

[56] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, et al. 2013. AS relationships, customer cones, and validation. In *Internet Measurement Conference (IMC)*. ACM.

[57] Alexander Marder and Jonathan M Smith. 2016. MAP-IT: Multipass Accurate Passive Inferences from Traceroute. In *Internet Measurement Conference (IMC)*. ACM.

[58] Microsoft. 2018. Azure ExpressRoute. https://azure.microsoft.com/en-us/services/expressroute/.

[59] Microsoft. 2018. ExpressRoute connectivity partners. https://azure.microsoft.com/en-us/services/expressroute/connectivity-partners/.

[60] Rich Miller. 2015. Regional Data Center Clusters Power Amazon's Cloud. https://datacenterfrontier.com/regional-data-center-clusters-power-amazons-cloud/.

[61] D. Moore, R. Periakaruppan, J. Donohoe, and k. claffy. 2000. Where in the world is netgeo.caida.org?. In *International Networking Conference (INET)*.

[62] Reza Motamedi, Bahador Yeganeh, Balakrishnan Chandrasekaran, Reza Rejaie, Bruce Maggs, and Walter Willinger. 2019. On Mapping the Interconnections in Today's Internet. *Transactions on Networking (ToN)* (2019).

[63] George Nomikos and Xenofontas Dimitropoulos. 2016. traIXroute: Detecting IXPs in traceroute paths. In *Passive and Active Measurement (PAM)*. Springer.

[64] Packet Clearing House. 2017. Routing Archive. https://www.pch.net.

[65] V. Padmanabhan and L. Subramanian. 2001. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *SIGCOMM*. ACM.

[66] Vern Edward Paxson. 1997. *Measurements and analysis of end-to-end Internet dynamics*. Ph.D. Dissertation. University of California, Berkeley.

[67] PeeringDB. 2017. Exchange Points List. https://peeringdb.com/.

[68] George Plaven. 2017. Amazon keeps building data centers in Umatilla, Morrow counties. http://www.eastoregonian.com/eo/local-news/20170317/amazon-keeps-building-data-centers-in-umatilla-morrow-counties.

[69] Amir H Rasti, Nazanin Magharei, Reza Rejaie, and Walter Willinger. 2010. Eyeball ASes: from geography to connectivity. In *Internet Measurement Conference (IMC)*. ACM.

[70] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 2011. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. *Selected Areas in Communications* (2011).

[71] Mario A Sánchez, John S Otto, Zachary S Bischof, David R Choffnes, Fabián E Bustamante, Balachander Krishnamurthy, and Walter Willinger. 2013. Dasu: Pushing experiments to the Internet's edge. In *NSDI*. USENIX.

[72] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. 2017. Engineering egress with edge fabric: Steering oceans of content to the world. In *SIGCOMM*. ACM.

[73] Justine Sherry, Ethan Katz-Bassett, Mary Pimenova, Harsha V Madhyastha, Thomas Anderson, and Arvind Krishnamurthy. 2010. Resolving IP Aliases with Prespecified Timestamps. In *Internet Measurement Conference (IMC)*. ACM.

[74] Rob Sherwood, Adam Bender, and Neil Spring. 2008. DisCarte: A Disjunctive Internet Cartographer. In *SIGCOMM CCR*. ACM.

[75] Neil Spring, Ratul Mahajan, and Thomas Anderson. 2003. The causes of path inflation. In *SIGCOMM*. ACM.

[76] Neil Spring, Ratul Mahajan, and David Wetherall. 2002. Measuring ISP Topologies with Rocketfuel. In *SIGCOMM CCR*. ACM.

[77] Neil T Spring, David Wetherall, and Thomas E Anderson. 2003. Scriptroute: A Public Internet Measurement Facility. In *Symposium on Internet Technologies and Systems*. USENIX.

[78] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang. 2011. Towards Street-Level Client-Independent IP Geolocation. In *NSDI*. USENIX.

[79] WikiLeaks. 2018. Amazon Atlas. https://wikileaks.org/amazon-atlas/.

[80] Mark Williams. 2016. Amazon's central Ohio data centers now open. http://www.dispatch.com/content/stories/business/2016/10/18/amazon-data-centers-in-central-ohio-now-open.html.

[81] Florian Wohlfart, Nikolaos Chatzis, Caglar Dabanoglu, Georg Carle, and Walter Willinger. 2018. Leveraging interconnections for performance: the serving infrastructure of a large CDN. In *SIGCOMM*. ACM.

[82] B. Wong, I. Stoyanov, and E. Sirer. 2007. Octant: A Comprehensive Framework for the Geolocation of Internet Hosts. In *NSDI*.

[83] Kuai Xu, Zhenhai Duan, Zhi-Li Zhang, and Jaideep Chandrashekar. 2004. On Properties of Internet Exchange Points and Their Impact on AS Topology and Relationship. In *Networking*. Springer.

[84] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, Ankur Jain, et al. 2017. Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *SIGCOMM*. ACM.