Project Description: CNS: Cloud Cartography: Measurement Capabilities for the Modern Internet

1 Introduction and Motivation

The growing deployment of low-latency and high-throughput applications, the upfront and maintenance costs of computing resources, and constantly evolving security threats make it increasingly complex and costly for organizations to host services and applications themselves. Public cloud providers, like Amazon AWS, Microsoft Azure, and Google Cloud Platform (GCP), ease that burden by allowing organizations to build and scale their applications on networks and hardware managed by the cloud provider. As applications shifted into the clouds, the Internet fundamentally changed from peer-to-peer to a cloud-centric model.

The importance of the clouds in the modern Internet necessitates understanding the paths between cloud applications and users to better inform public policy and network operations. Here, we propose an ambitious effort to directly observe and interpret these cloud application paths at the router level, using cloud virtual machines (VMs) to embed measurement infrastructure in the same networks and data centers that house cloud applications. We structure our research in two tasks: extending our current topology inference methodology to interpret traceroute path measurements conducted *from* cloud VMs *to* end-hosts outside the cloud, and accurate interpretation of individual traceroute paths *from* end-hosts *toward* the cloud. Paths between two endpoints can differ in each direction due to traffic engineering and best-path selection, and accomplishing these two tasks will help network operators understand the paths between their networks and the cloud, and to diagnose problems along those paths. In the future, our solutions will unlock new avenues of research vital to understanding the reliability, robustness, and failures of the modern Internet, and provide the type of information valuable to regulators, businesses, and network operators.

Our first task focuses on revealing and interpreting cloud wide area network (WAN) paths. Recently, we reported on techniques to map routers visible in traceroute collections to network operators, and to identify the points of interconnection between the networks [1–3]. By extending our techniques to cloud networks, we can create maps containing the paths from each cloud region to every corner of the Internet, along with the network operator for every observed router IP address. We will not attempt to guess or approximate the paths that clouds use to reach end-hosts; instead, we will observe the router paths used by public cloud WANs through comprehensive probing from our VMs. This information can help network operators diagnose problems, plan network improvements, and choose primary or backup providers. In the future, we expect that the techniques and annotated topology maps that we generate will enable third-party verification and comprehensive analysis of cloud WANs, such as detecting the location of congestion or packet loss between clouds and users.

Our second task will use the road-map of cloud WAN paths from Task 1 to similarly interpret individual traceroute paths in the reverse direction, from end-hosts to the cloud. Interpreting the networks visible in an individual path measurement from an arbitrary end-host is notoriously difficult due to the lack of constraints, even without identifying the points of network interconnections. We plan to use our preprocessed maps to provide the constraints needed to interpret traceroute toward large public clouds. Our partial solution to this decades-old problem would allow network operators to understand the paths that they take to third-party cloud applications, and to quickly diagnose and locate problems between their networks and cloud-hosted applications. This work would also enable a profound new capability that could immediately enable practical measurements from mobile devices to cloud applications and complement our measure-

ments from cloud VMs by providing visibility into the paths from users to the clouds. Conceivably, this capability could eventually allow even a non-technical user to determine sources of Internet frustration, such as the reason video communication freezes and lags.

2 Related Work and Background

Problems with traceroute interpretation date back to the original RFC specification [4] that routers respond to TTL-expiring packets with the address of the responding interface, at odds with the conventional traceroute interpretation that routers respond with their inbound address [5,6] either due to assumptions of symmetric routing or router implementations [7]. We review the relevant prior work from the following subfields: alias resolution, traceroute topology inference, and cloud network measurement.

Alias Resolution Alias resolution groups the router interface IP addresses seen in traceroute according to their physical router, providing valuable constraints to traceroute inference. Until recently, alias resolution primarily either exploited implementations of IP with active probing or analyzed graphs of traceroute paths, with graph analysis techniques [8–10] generally performing less accurately. Two reliable active techniques, Mercator [11] and iffinder [12], exploit certain router implementations that report the transmitting interface address when originating Destination Unreachable packets, indicating that the probed and transmitting interface addresses belong to the same router, although many routers either report the probed address or do not respond to the probes. UAv6 [13] extends this idea, sending probes to the network and broadcast addresses in IPv4 /30 and IPv6 /126 subnets.

Other active probing approaches draw inferences from the IPv4 IP-ID field, used to aid reassembly of fragmented packets, that some routers populate using a single counter for all interfaces. The Rocketfuel [6] component Ally compares pairs of addresses to see if the IP-IDs increase at similar rates, and RadarGun [14] removes the need to compare each pair of addresses separately, sampling each IP address a fixed number of times. MIDAR [15], the current state-of-the-art, scales the RadarGun approach to millions of addresses, and ensures that the IP-IDs of inferred aliases form a monotonically increasing sequence. IPv6 lacks the IP-ID field in the standard IP packet header, so Speedtrap [16] attempts to induce fragmented ICMP Echo Replies with IP-IDs, but some routers do not fragment packets in IPv6. In general, the future of IP-ID-based alias resolution is uncertain, as current IETF recommendations advise against setting the IP-ID in IPv4 packets outside of packet fragmentation [17].

Recently, two validated techniques moved beyond exploiting router-specific implementations of IP, with applicability to both IPv4 and IPv6. hoiho [18], uses machine learning to automatically generate regular expressions to extract router identifier labels embedded in DNS hostnames. Earlier this year, we reported on APPLE [19], which starts with simple graph analysis but uses the Time to Live (TTL) and round trip times (RTTs) of replies to pings from topologically distributed vantage points to refine the potential router alias groups. We showed that reply TTLs and RTTs can help discriminate between physical routers. In Task 2 we combine APPLE with MIDAR to match previously unseen addresses with addresses visible from cloud VMs.

Traceroute Topology Inference Early traceroute topology inference work focused on converting traceroute IP address paths into AS paths. The canonical approach mapped each IP address in the path to the AS that originated the longest matching prefix into BGP. The risk of this approach is that a router operated by one network might respond to a traceroute probe with a source IP address belonging to a different network [7,20,21]. Mao *et al.* in particular noticed the promise of this work for network diagnosis. Their "AS traceroute" [22] used correlated BGP and traceroute views

from the same VP, DNS names, and WHOIS data to perform IP-AS mappings, later improving them further using dynamic programming, although only at a /24 address granularity [23].

Two routers require addresses in the same IP subnet to interconnect, so mappings at any granularity coarser than individual IP addresses are inherently inaccurate for addresses used at network interconnections. Recognizing that individual traceroutes contain insufficient constraints for accurate interpretation, Huffaker [24] *et al.* developed and validated techniques that added outside constraints using alias resolution, but the incompleteness of alias resolution techniques severely limited their accuracy. In 2016, we reported on MAP-IT, which sidestepped the limitations of external alias resolution by drawing router alias constraints from the traceroute paths, and inferring network operators through iterative constraint satisfaction. Concurrent with MAP-IT, a separate group of researchers developed bdrmap, focusing on the more narrow problem of mapping the border of a single network hosting a single traceroute vantage point. Two years later, we led the effort to improve on the approaches in MAP-IT and bdrmap, creating a novel methodology we call bdrmapIT. In creating bdrmapIT, we adapted MAP-IT's iterative constraint satisfaction to incorporate new types of constraints, such as alias resolution and AS business relationships, and adapted and generalized many of bdrmap's heuristics to account for the difficulties of synthesizing information from multiple traceroute vantage points (VPs), and inferring router operators and network interconnections multiple AS hops from a vantage point. Across several studies we validated bdrmapIT against ground truth provided by ISP network operators, IXP public peering address assignments that networks report to PeeringDB, and AS numbers embedded in DNS hostnames.

Cloud Wide-Area Networks There has been substantial prior work on data center networks [25–32], but few that measure the external connectivity of data centers or cloud networks. Some studies recognized the fundamental traceroute limitation of only revealing paths *from* the VP. Katz-Basset *et al.* [33] proposed a technique to approximate the reverse-path from a destination to the VP, and Cunha *et al.* [34] attempted to predict paths of interest, each with some success. Rather than approximating cloud paths to public VPs, or predict cloud paths from largely irrelevant traceroute collections, we propose to reveal and interpret cloud network paths to the entire public Internet by conducting traceroutes from inside public cloud data centers.

Yeganeh *et al.* [35] conducted traceroutes from AWS to every /24 to reveal interconnected networks, using a new unvalidated approach to infer network interconnections. In subsequent work [36], they compared the quality of service of default interconnections between cloud networks and third-party transit between clouds, switching to bdrmapIT to perform interconnection IP addresses inferences. Relatedly, Arnold *et al.* [37] analyzed latency from distributed vantage points to cloud VMs using cloud network WANs and public Internet transit, showing that WANs generally reduced latency compared to the public Internet. In later work, Arnold *et al.* [38] inferred directly connected networks from traceroute paths by converting traceroute IP addresses to ASes using longest-matching prefix in BGP route announcements and IXP participant IP addresses recorded in PeeringDB [39]. They then augmented the AS-level connectivity graph in CAIDA's AS Relationship dataset [40] with peer relationships between each cloud and the newly inferred neighbors, using the graph to estimate that clouds can avoid their transit providers listed in CAIDA's AS Relationship dataset to reach 76% of the Internet networks. They validated their neighbor inferences with feedback from Azure and GCP, with 11%–15% false neighbor inferences. Assuming nearly perfect accuracy for IXP participant addresses in PeeringDB, these false neighbor inferences almost entirely result from false private interconnection inferences.

Rather than use unvalidated AS interconnection inference techniques, we plan to use the previously validated bdrmapIT tool to infer interconnections between cloud public WANs and their

neighbors, and between downstream networks. We also plan to perform additional validation to understand bdrmapIT's accuracy for cloud networks. Finally, while Arnold *et al.* speculated how clouds *could* reach other ASes [38], we report how clouds currently *do* reach other networks.

3 Task 1: Reveal and Interpret Router-Paths From Clouds

Prior AS-traceroute efforts [21,23] attempted to identify the AS-level path from a single traceroute path by aligning it with known BGP AS paths. Neither approach validated their technique, and reliably solving this problem proved difficult. Identifying the AS interconnection IP addresses in an individual traceroute path is impractical without substantial supplemental information, since a single path lacks the necessary information to constrain inference.

We propose to solve this problem specifically for the major public cloud providers, such as AWS, Azure, and GCP. We do not expect to overcome the fundamental problem of insufficient constraints for an individual traceroute path, but instead intend to leverage comprehensive pre-processed maps of the routers and network interconnection visible from vantage points (VPs) inside cloud networks. Placing VPs inside cloud networks allows us to observe the router-paths that clouds use to reach every corner of the Internet over their public WANs. To identify how the cloud reaches an end-host—at that very moment—we can issue a traceroute from the cloud to that host, and interpret it using the preprocessed maps.

This approach, discovering and interpreting the topology from a set of VP co-located with popular applications was theoretically possible before, but applications and services were often hosted in private data centers. The expense of hosting VPs in those data centers, the volume of data centers hosting these applications, and the fact that deployment in a data center often acted as an extension of an organization's network combined to make such a solution impractical and unscaleable. The only practical approach was to guess the paths that applications might use to reach their users, as with the Sybil [34] technique. Guessing paths cannot provide the reliable information that network operators need for diagnostics and planning.

Today, with applications and services centralizing in the cloud, we have the ability to co-locate measurement VPs and applications in public cloud data centers. Measurements from these VPs have access to the same public WAN used by application traffic to reach users. Limiting the scope of our problem to a handful of networks helps make this problem tractable, while still providing significant benefit to network operators. In this task, we propose to (1) comprehensively discover the router-level paths from public cloud providers to the rest of the Internet; and (2) generate maps from the paths annotated with AS operators for each of the observed routers and points of interconnection between networks; and (3) use our maps to reliably interpret individual traceroute paths from cloud networks to end-hosts.

3.1 Research Questions: Revealing and Interpreting the Cloud-Centric Internet

The first question we plan to answer is how accurate are current techniques for inferring AS operators for observed routers and for identifying cloud network interconnections? Inferring router operators and interconnections from traceroute paths is notoriously difficult, and while our bdrmapIT tool is the first empirically accurate solution to this problem, cloud networks are fundamentally different from the ISPs used for bdrmapIT's validation. Relatedly, the extensive traceroute probing required to generate our topology maps potentially introduces significant noise in the form of traceroute path corruptions. How can we comprehensively reveal the topology while retaining signal but limiting noise?

The dataset of router operators and AS interconnections can also help us learn about the cloud-centric Internet. Reliable cloud connectivity is vital to daily life in the United States, but are there certain states, cities, or zip codes where a public cloud lacks the path diversity for robust connectivity, and how does cloud path diversity in the US compare to the rest of the world? Furthermore, as the Internet shifts from peer-to-peer to a cloud-centric model, the role of IXPs shifted as well to facilitate cloud connectivity. A natural question is how do clouds incorporate IXPs into their interconnection strategy, and does the strategy differ depending on the geographic region? Of particular importance to enterprises with multi-cloud deployments, where do the public clouds interconnect, does traffic between two cloud networks ever touch the public Internet, and do clouds select different interconnections depending on destination regions? These questions will help guide our research and lay the groundwork for future work.

3.2 Approach and Challenges

Interpreting an individual traceroute to infer the ASes traversed and identify the network interconnections is an intractable challenge due to the lack of constraints. We propose sidestep this problem by collecting and preprocessing constraints *before* interpreting individual traceroutes from cloud networks toward end-hosts. Our approach will use comprehensive probing to every /24 covered by a prefix in BGP route announcements, covering every IP prefix at the smallest granularity typically announced into BGP. We plan to generate new road-map from the cloud to every /24, annotated with infer the AS operator for each router observed in those traceroute paths, and use those inferences to identify the interconnections between networks. We observe the following three challenges that our solution must overcome: (1) limited feedback from cloud network operators makes it difficult to refine our techniques; (2) path changes corrupt traceroute paths; and (3) GCP obscures traceroute paths with probe TTL manipulation.

Limited Feedback From Cloud Network Operators Our analysis relies on bdrmapIT AS operator inferences to identify cloud interconnections and neighbors, so we first need to validate bdrmapIT's inferences on traceroutes from cloud VMs to gain confidence in its efficacy and look for opportunities to improve our techniques. We assume that validation of transit interconnection inferences might not translate to cloud interconnection inferences. Initial bdrmapIT evaluations used CAIDA's Ark traceroutes and ground truth from ISP operators, and later experiments also validated bdrmapIT against pseudo ground truth derived from ISP DNS hostnames [3, 41, 42]. Traceroutes from CAIDA's Ark VPs mostly reveal transit interconnections—those between providers and customers—so transit interconnections dominate their reported accuracy. Clouds primarily peer with other networks, and we expect that their peering interconnections vastly outnumber their transit interconnections. Importantly, bdrmapIT leverages the industry convention that transit providers supply the IP subnets for interconnection with customers, but no known convention exists for peering interconnections [43]. To date, no study has evaluated bdrmapIT's accuracy using traceroutes that originate in the cloud.

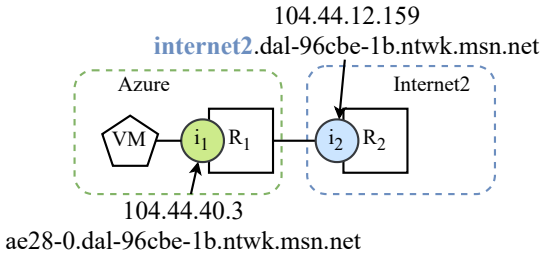


Figure 1: The `internet2` tag indicates that `104.44.12.159` belongs to a router operated by Internet2. We can use this as preliminary validation for bdrmapIT's router operator inferences from cloud VM traceroutes.

The only comprehensive and definitive validation requires cooperation from network opera-

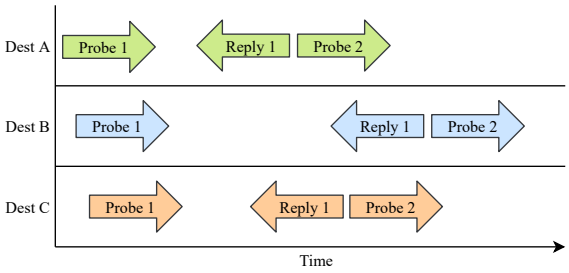


Figure 2: Scamper increases efficiency by parallelizing traceroute probing across destinations, but a path change can corrupt all active traceroute paths.

Subsequent	Appearances	Percentage
110.164.0.112	5811	48.55%
110.164.0.188	5805	48.50%
27.111.228.76	351	2.93%
104.44.7.12	**1**	**0.01%**

Figure 3: Addresses following `104.44.13.3` (`jastel.sge-96cbe-1b.ntwk.msn.net`). The Azure address occurs only once, likely due to traceroute path corruption.

tors at the public clouds. To avoid giving third-party researchers potentially confidential information about their interconnections, network operators often only provide summary numbers, making it difficult to improve our inferences. This process also consumes network operator time, allowing for few iterations of improvement and new validation. We hope to validate our approach with ground truth from network operators at cloud networks, but first plan to use pseudo ground truth to make the process of refining our methodologies more efficient.

Initially, we will use DNS hostnames to provide pseudo ground truth for our interconnection inferences, allowing us to quickly refine our technique. In a preliminary study using three traceroute runs from every Azure region to an IPv4 address in every routed /24, we successfully resolved hostnames for 59.5% of the 5749 Azure IP addresses seen in our initial traceroutes, and they often indicate when the Azure address belongs a router operated by a neighboring network (Fig. 1); e.g., `internet2.dal-96cbe-1b.ntwk.msn.net` indicates that `104.44.12.159` belongs to an Internet2 router interconnected with Azure. We can extract the interconnection tags and manually interpret the interconnected network referenced by the tag to create router-level validation. We cannot extend this type of DNS hostname validation to AWS and GCP, since we resolved few hostnames for their IP address seen in traceroute, and found no interconnection tags in resolved hostnames.

Path Changes Can Corrupt Traceroute Paths At a minimum, we plan to conduct traceroutes from every region of each cloud to every IPv4 /24 prefix covered by a route announcement in public BGP collections. Discovering topology from every cloud region helps capture any differences between regions. We chose the /24 granularities as networks typically filter out longer prefix announcements. Probing to the smallest routable unit typically allowed in BGP helps capture interdomain traffic engineering policies.

To avoid stale paths we plan to conduct traceroutes to every /24 daily, requiring an efficient method for revealing traceroute paths. Conventional traceroute probing waits for the response to the probe with TTL i before sending the probe with TTL $i + 1$, and topology discovery tools like Scamper [44] parallelize across traceroute destinations to increase efficiency (Fig. 2). This concurrency enables rapid path discovery, necessary for temporally coherent snapshots of cloud topologies, but a path change can corrupt any of the traceroutes active at any given time. We expect that topology discovery tools that parallelize across all destinations simultaneously, like Yarrp [45] and FlashRoute [46], are even more prone to path change corruptions. In our preliminary traceroutes, collected by Scamper, we found 63 (14.6%) Azure addresses with an interconnection tag in their hostnames followed by at least one Azure address in a traceroute. These interconnection addresses are on routers operated by neighboring networks, so an uncorrupted traceroute path should not

contain a subsequent Azure address.

We plan to address the problem of traceroute path corruptions due to path changes with two techniques. First, we plan to experiment with new probing methods that send few traceroutes concurrently, but still rapidly reveal the visible topology. Rather than wait for responses, these traceroute techniques could probe at a fixed rate; e.g., probing at 5000 packets per second (PPS) could complete an IPv4 topology discovery run to all 11.5 M routed /24s in less than 21 hours. An initial test sending a single traceroute at once found only one traceroute where an Azure address followed an address with an interconnection tag in its hostname.

While our first approach might substantially reduce errors due to path changes from corrupting traceroute paths, the fundamental problem that increased probing increases noise remains. This paradox, that more traceroutes can lead to worse inferences, occurs because a set of VPs can only reveal a finite amount of topology. As they approach that point, traceroutes either reveal no new information or introduce noise in the form of path corruptions. We expect that noise is mostly anomalous, and should not repeat often relative to paths that reflect the topology, making it possible to detect and prune some corrupt paths. In Fig. 3, the Azure address `104.44.7.12` follows an address with an interconnection tag in its hostname in only one of 11,968 traceroute paths that revealed the interconnections address, providing an opportunity to prune the anomalous path.

GCP Manipulates Probe TTLs Another challenge for our analysis is that GCP inflates the TTL values of traceroute probes after they leave VMs such that the hop #1 traceroute address belongs to a later router in that path, rather than to the first router hop [47], and traceroute paths might not begin in GCP's WAN. This practice of rewriting probe TTLs violates a core traceroute assumption that hop #1 corresponds to the first router probed, and likely caused researchers to incorrectly conclude that GCP routers do not respond to traceroute [48], or that hop #1 is a router just past the GCP border [36]. While AWS and Azure also hide early router hops in traceroute paths, they do so by preventing responses to the VMs, rather than rewriting the TTL, and their paths all appear to start inside their WANs.

Dest: 158.130.69.163
1 128.91.238.217 [UPenn]
2 128.91.48.6 [UPenn]

(a) LA to UPenn.

Dest: 146.97.33.5
1 216.239.59.1 [Google]
2 172.253.65.167 [Google]
3 209.85.143.66 [Google]
4 108.170.246.168 [Google]
5 *
6 146.97.33.62 [JANET]
7 146.97.33.5 [JANET]

(b) LA to JANET.

Dest: 158.130.69.163
1 209.85.253.197 [Google]
2 172.253.65.176 [Google]
3 108.170.227.150 [Google]
4 108.170.248.11 [Google]
5 162.252.69.196 [Internet2]
6 *
7 128.91.238.218 [Internet2]
8 128.91.238.217 [UPenn]
9 128.91.48.6 [UPenn]

(c) Belgium to UPenn.

Figure 4: A traceroute from GCP Los Angeles to the University of Pennsylvania (UPenn) revealed no GCP IP addresses (a), but traceroutes from Los Angeles to JANET in the UK (b), and Belgium to UPenn (c), each revealed GCP addresses.

Fig. 4a shows the GCP TTL inflation with a traceroute from a VM in Los Angeles, where hop #1 reported an address that router configurations from Internet2 show belong to a University of Pennsylvania (UPenn) router, despite no direct interconnection between GCP and UPenn. The inflated TTL caused probes to expire only after reaching UPenn. Traceroutes from other GCP VMs to the same UPenn destination, such as in the Belgium region, exposed apparent GCP internal IP addresses, only reaching UPenn at hop 8 (Fig. 4c). All of our VMs use GCP's premium network tier, but not all revealed internal GCP addresses, contradicting reported behavior that only GCP's standard tier inflates traceroute TTLs [37]. Our ability to observe internal GCP addresses from the Belgium VM toward UPenn, and from the VM in Los Angeles toward JANET in the UK (Fig. 4b), suggests that the opportunity to view internal and interconnection GCP addresses depends on the combination of GCP region and traceroute destination.

4 Task 2: Interpreting Observed Paths From End-Hosts to Clouds

In Task 1, we plan to conduct extensive traceroute probing to interpret individual traceroutes from the cloud. The current importance of cloud paths, and the applicability of our generated maps to all paths traversing cloud public WANs, makes our solution worthwhile despite the high probing and time cost. This reasoning does not generalize to edge networks like enterprises and universities, as they rarely use the vast majority of the paths that our comprehensive probing reveals. With applications centralizing in public clouds, we expect paths to public clouds are among the most important paths to these edge networks.

Rather than ask edge networks to repeat our methodology in their own networks to understand a handful of important paths, in Task 2 we apply our solution from Task 1 to the reverse paths from end-users to public clouds, allowing network operators to understand how traffic reaches cloud applications from their networks. Our goal is not to guess the paths that a network might take to a cloud region. Instead, given an actual traceroute path from an edge host to a cloud destination, we plan to interpret the observed path mapping each hop to an AS and identify network interconnections. This information can help network operators diagnose unexpectedly high latency between their network and specific cloud applications, or locate the network responsible for packet drops.

4.1 Research Questions: Plotting Individual Paths on Topology Maps

This task is not straightforward, since IP addresses on these reverse paths will not appear in our preprocessed maps, unlike the paths observed in Task 1. A central question for this task is how to fit previously unseen IP addresses on an existing router graph? Alias resolution, the process of inferring which router interface IP address in traceroute paths belong to the same router, is a natural solution to this problem. Current approaches to alias resolution require extensive and time-consuming probing, and must test addresses contemporaneously. Can we adapt alias resolution techniques to identify addresses already in the graph that belong to the same routers as the handful of previously unseen addresses in a new traceroute path? Furthermore, routers interconnect over IP, so interconnected routers must have an address in the same IP subnet. How should we leverage point-to-point network interconnections between two routers to identify the interconnections, in the absence of alias resolution, but avoid errors caused by off-path address misinformation? Off-path addresses remain an unsolved problem in traceroute interpretation, so we will need to recognize suspicious information and discard it.

4.2 Approach and Challenges

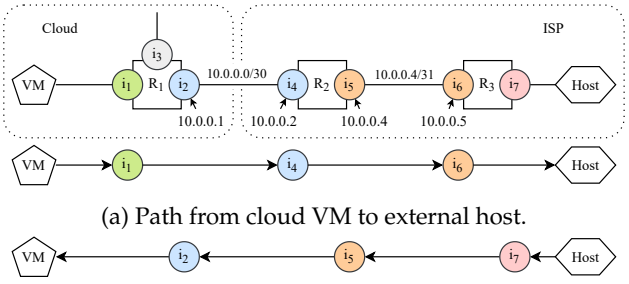
Our approach in this task is to fit traceroute paths from end-hosts toward cloud destinations to the annotated maps generated in Task 1. Challenges stem from the fact that addresses in traceroute paths *toward* the cloud will not typically appear in paths *from* the cloud. As a result, we need heuristics to identify the routers in our map represented by the addresses in the traceroute path.

In Fig. 5, path (a) shows a traceroute from the cloud VM to an end-host. Routers usually respond with the address of the interface that received the traceroute probe, so each router responds with an address from the cloud's side of the router. Conversely, path (b) contains addresses on the host's side of the routers. Despite traversing the same routers, the traceroute paths contain entirely different addresses.

Two techniques can help fit the traceroute from the end-host in path (b) to the routers seen in path (a): alias resolution and subnet matching. Importantly, neither technique requires the

symmetric paths in Fig. 5, only that each router appears in the map.

Reducing Alias Resolution Search Space Alias resolution attempts to group router interface IP addresses observed in traceroute paths according to their physical router, and is a natural fit when trying to match IP addresses in a path toward a cloud to IP addresses in the maps created in Task 1. The most accurate alias resolution technique, MIDAR, compares IP-ID slopes across multiple ping responses to identify IP addresses on the same router. A router's IP-ID slope is not a fixed property, so accurately comparing slopes relies on probing each address within minutes to hours.



(a) Path from cloud VM to external host.

(b) Path from external host to the cloud VM.

Figure 5: Traceroutes from cloud VMs (a) reveal different addresses than traceroutes to clouds (b).

Pinging the millions of addresses in our map to identify aliases of addresses in the new traceroute path would take impractically long, so one challenge is reducing the search space. Recently, we showed that router responses to pings from a VP typically have similar reply TTLs and RTTs regardless of the response source address, and that the combination of reply TTL and RTT can discriminate between routers [19]. The reply TTL here refers to the remaining TTL value in the response IP header when the response reaches the VP. Unlike the IP-ID slope, we expect that both the reply TTL and the RTT for a router to a VP remains roughly the same for days or weeks.

We plan to use reply TTLs and RTTs to our cloud VPs to sufficiently reduce the search space to enable MIDAR-style alias resolution for new traceroute paths. Traceroute replies sometimes use a different starting TTL than ping replies, so after creating the map, we ping every address to determine the average reply TTL and RTT to each cloud region. Our preliminary Azure probing revealed 3,609,132 unique ICMP Time Exceeded and Destination Unreachable addresses, and pinging each address three times at 5000 PPS takes around 36 minutes. When presented with a traceroute path toward the cloud, we only need to compare IP-ID slopes for addresses with similar reply TTLs and RTTs as the addresses in the path.

Off-Path Addresses Confound IP Subnet Matching Alias resolution does not always succeed, as some routers and interface addresses are unresponsive to ping. For the remaining address in the new traceroute path, we can attempt to infer the point-to-point subnet for each address. In IPv4, the point-to-point subnets either include four addresses (/30) or two addresses (/31). We recently presented a technique to infer if a router IP address belongs to a two- or four-address subnet that checks if the network or broadcast address in the potential /30 responded to traceroute or ping, indicating that the address belongs to a /31 [41].

After determining the subnet, we can identify the AS operator for its router using our map. In Fig. 5b, the traceroute path to the cloud revealed `10.0.0.1` from a /30 subnet. The other address in that subnet appeared in a cloud traceroute immediately subsequent to router R_1. We can use the AS operator annotation for R_1 to infer that `10.0.0.1` belongs to a router operated by the cloud network, and interconnects the cloud network with the ISP.

Off-path addresses in traceroute paths, where a router responds with an IP address belonging to an interface that did not receive the probe, can lead to errors when inferring router operators from IP subnet matching. One strategy we can employ is to mitigate the problems caused by off-path addresses using knowledge of directly connected ASes learned from BGP AS paths and from the cloud interconnections discovered in Task 1. When IP subnet matching separates routers

operated by the same network, or two different networks known to interconnect, we can discard the likely erroneous inference.

References

[1] A. Marder and J. M. Smith, "MAP-IT: Multipass accurate passive inferences from traceroute," in *Proceedings of the 2016 Internet Measurement Conference*, pp. 397–411, ACM, 2016.

[2] A. Marder, M. Luckie, A. Dhamdhere, B. Huffaker, kc claffy, and J. M. Smith, "Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale," in *IMC*, 2018.

[3] M. Luckie, A. Marder, M. Fletcher, B. Huffaker, and kc claffy, "Learning to extract and use ASNs in hostnames," in *Proceedings of the 2020 Internet Measurement Conference*, 2020.

[4] J. Postel, "RFC 791: Internet Protocol," tech. rep., Internet Engineering Task Force, 1981.

[5] R. A. Steenbergen, "A practical guide to (correctly) troubleshooting with traceroute," *NANOG*, 2009.

[6] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," *ACM SIGCOMM Computer Communication Review*, 2002.

[7] L. D. Amini, A. Shaikh, and H. G. Schulzrinne, "Issues with inferring Internet topological attributes," in *Internet Performance and Control of Network Systems III*, 2002.

[8] N. Spring, M. Dontcheva, M. Rodrig, and D. Wetherall, "How to resolve IP aliases," Tech. Rep. UW-CSE-TR 04–05–04, University of Washington, 2004.

[9] M. H. Gunes and K. Sarac, "Resolving IP aliases in building traceroute-based Internet maps," *IEEE/ACM Transactions on Networking*, 2009.

[10] K. Keys, "Internet-scale IP alias resolution techniques," *ACM SIGCOMM Computer Communication Review (CCR)*, 2010.

[11] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000.

[12] K. Keys, "iffinder." `https://www.caida.org/tools/measurement/iffinder/`.

[13] R. Padmanabhan, Z. Li, D. Levin, and N. Spring, "UAv6: Alias resolution in IPv6 using unused addresses," in *PAM*, 2015.

[14] A. Bender, R. Sherwood, and N. Spring, "Fixing ally's growing pains with velocity modeling," in *IMC*, 2008.

[15] K. Keys, Y. Hyun, M. Luckie, and K. Claffy, "Internet-scale IPv4 alias resolution with MIDAR," *IEEE/ACM Transactions on Networking*, 2013.

[16] M. Luckie, R. Beverly, W. Brinkmeyer, and k. claffy, "Speedtrap: internet-scale ipv6 alias resolution," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 119–126, 2013.

[17] J. Touch, "RFC 6864: Updated specification of the ipv4 id field," tech. rep., Internet Engineering Task Force, Feb 2013.

[18] M. Luckie, B. Huffaker, and k. claffy, "Learning regexes to extract router names from hostnames," in *Proceedings of the Internet Measurement Conference*, pp. 337–350, 2019.

[19] A. Marder, "APPLE: Alias pruning by path length estimation," in *International Conference on Passive and Active Network Measurement*, pp. 249–263, Springer, 2020.

[20] Y. Zhang, R. Oliveira, H. Zhang, and L. Zhang, "Quantifying the pitfalls of traceroute in as connectivity inference," in *International Conference on Passive and Active Network Measurement*, 2010.

[21] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao, "Where the sidewalk ends: Extending the Internet AS graph using traceroutes from P2P users," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pp. 217–228, ACM, 2009.

[22] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate as-level traceroute tool," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 365–378, ACM, 2003.

[23] Z. M. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz, "Scalable and accurate identification of as-level forwarding paths," in *IEEE INFOCOM*, vol. 3, pp. 1605–1615, Citeseer, 2004.

[24] B. Huffaker, A. Dhamdhere, M. Fomenkov, and kc claffy, "Toward topology dualism: improving the accuracy of as annotations for routers," in *International Conference on Passive and Active Network Measurement*, pp. 101–110, Springer, 2010.

[25] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. USENIX NSDI*, Apr. 2010.

[26] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. ACM IMC*, Nov. 2010.

[27] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine grained traffic engineering for data centers," in *Proc. ACM CoNEXT*, Dec. 2011.

[28] C. Delimitrou, S. Sankar, A. Kansal, and C. Kozyrakis, "ECHO: Recreating network traffic maps for datacenters with tens of thousands of servers," in *Proc. IEEE International Symposium on Workload Characterization*, Nov. 2012.

[29] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM*, Aug. 2009.

[30] S. Kandula, J. Padhye, and P. Bahl, "Flyways to de-congest data center networks," in *Proc. ACM HotNets*, Oct. 2009.

[31] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: Measurements & analysis," in *Proc. ACM IMC*, Nov. 2009.

[32] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proceedings of the ACM SIGCOMM Conference*, (London, England), Aug. 2015.

[33] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. Van Wesep, T. E. Anderson, and A. Krishnamurthy, "Reverse traceroute.," in *NSDI*, vol. 10, pp. 219–234, 2010.

[34] Í. Cunha, P. Marchetta, M. Calder, Y.-C. Chiu, B. V. Machado, A. Pescapè, V. Giotsas, H. V. Madhyastha, and E. Katz-Bassett, "Sibyl: a practical internet route oracle," in *NSDI*, 2016.

[35] B. Yeganeh, R. Durairajan, R. Rejaie, and W. Willinger, "How cloud traffic goes hiding: A study of amazon's peering fabric," in *IMC*, pp. 202–216, 2019.

[36] B. Yeganeh, R. Durairajan, R. Rejaie, and W. Willinger, "A first comparative characterization of multi-cloud connectivity in today's internet," in *PAM*, pp. 193–210, Springer, 2020.

[37] T. Arnold, E. Gürmeriçliler, G. Essig, A. Gupta, M. Calder, V. Giotsas, and E. Katz-Bassett, "(how much) does a private wan improve cloud performance?," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 79–88, IEEE, 2020.

[38] T. Arnold, J. He, W. Jiang, M. Calder, I. Cunha, V. Giotsas, and E. Katz-Basset, "Cloud provider connectivity in the flat internet," *IMC*, 2020.

[39] "PeeringDB." `https://peeringdb.com/`.

[40] "The CAIDA AS relationships dataset." `https://www.caida.org/data/as-relationships/`.

[41] A. Marder, M. Luckie, B. Huffaker, and kc claffy, "vrfinder: Finding outbound addresses in traceroute," in *SIGMETRICS*, 2020.

[42] A. Marder, *Sharp Snapshots of the Internet's Graph with HONE*. PhD thesis, University of Pennsylvania, 2019.

[43] M. Luckie, A. Dhamdhere, D. Clark, B. Huffaker, and K. Claffy, "Challenges in inferring internet interdomain congestion," in *IMC*, 2014.
[44] M. Luckie, "Scamper: a scalable and extensible packet prober for active measurement of the Internet," in *IMC*, 2010.
[45] R. Beverly, "Yarrp'ing the internet: Randomized high-speed active topology discovery," in *IMC*, pp. 413–420, 2016.
[46] Y. Huang, M. Rabinovich, and R. Al-Dalky, "Flashroute: Efficient traceroute on a massive scale," in *Proceedings of the ACM Internet Measurement Conference*, pp. 443–455, 2020.
[47] "VPC network overview." `https://cloud.google.com/vpc/docs/vpc`, May 2020.
[48] O. Haq, M. Raja, and F. R. Dogar, "Measuring and improving the reliability of wide-area cloud paths," in *WWW*, pp. 253–262, 2017.