Tiered Cloud Routing: Methodology, Latency, and Improvement

SHIHAN LIN, Duke University, USA
YI ZHOU, Duke University, USA
XIAO ZHANG*, Cisco ThousandEyes, USA
TODD ARNOLD†, U.S. Military Academy, USA
RAMESH GOVINDAN, University of Southern California, USA
XIAOWEI YANG, Duke University, USA

Large cloud providers including AWS, Azure, and Google Cloud offer two tiers of network services to their customers: one class uses the providers' private wide area networks (WAN-transit) to carry a customer's traffic as much as possible, and the other uses the public internet (inet-transit). Little is known about how each cloud provider configures its network to offer different transit services, how well these services work, and whether the quality of those services can be further improved. In this work, we conduct a large-scale study to answer these questions. Using RIPE Atlas probes as vantage points, we explore how traffic enters and leaves each cloud's WAN. In addition, we measure the access latency of the WAN-transit and the inet-transit service of each cloud and compare it with that of an emulated performance-based routing strategy. Our study shows that despite the cloud providers' intention to carry customers' traffic on its WAN to the maximum extent possible, for about 12% (Azure) and 13% (Google) of our vantage points, traffic exits the cloud WAN early at cloud edges more than 5000 km away from the vantage points' nearest cloud edges. In contrast, more than 84% (AWS), 78% (Azure), and 81% (Google) of vantage points enter a cloud WAN within a 500 km radius of their respective locations. Moreover, we find that cloud providers employ different routing strategies to implement the inet-transit service, leading to transit policies that may deviate from their advertised service descriptions. Finally, we find that a performance-based routing strategy can significantly reduce latencies in all three cloud providers for 4% to 85% of vantage point and cloud region pairs.

CCS Concepts: • **Networks** → **Network measurement**; **Public Internet**; **Routing protocols**.

Additional Key Words and Phrases: BGP, Wide Area Network, Internet routing, Cloud routing

ACM Reference Format:
Shihan Lin, Yi Zhou, Xiao Zhang, Todd Arnold, Ramesh Govindan, and Xiaowei Yang. 2025. Tiered Cloud Routing: Methodology, Latency, and Improvement. *Proc. ACM Meas. Anal. Comput. Syst.* 9, 1, Article 12 (March 2025), 41 pages. https://doi.org/10.1145/3711705

1 INTRODUCTION

In recent years, large cloud service providers such as Amazon Web Services (AWS) [7], Microsoft Azure [23], and Google Cloud [15] began providing two-tiered network services to their customers.

*Xiao Zhang was with Duke University at the time this work was conducted. He is now with Cisco ThousandEyes.
†The views expressed herein are those of the authors and do not reflect the position of the US Military Academy, Department of the Army, or Department of Defense.

Authors' addresses: Shihan Lin, Duke University, Durham, USA; Yi Zhou, Duke University, Durham, USA; Xiao Zhang, Cisco ThousandEyes, Raleigh, USA; Todd Arnold, U.S. Military Academy, West Point, USA; Ramesh Govindan, University of Southern California, Los Angeles, USA; Xiaowei Yang, Duke University, Durham, USA.

ACM 2476-1249/2025/3-ART12
https://doi.org/10.1145/3711705

The *WAN-transit* service uses a cloud provider's private wide-area network (WAN) to carry cloud traffic. The lower tier *inet-transit* service carries cloud traffic over the public Internet. In theory, the former offers better performance and reliability, since the private WANs are better provisioned and maintained.

To achieve this, cloud providers use inter-domain route advertisement strategies designed to steer traffic between users and the cloud region[1] hosting a service. For example, to provide inet-transit service, a cloud provider might advertise a Border Gateway Protocol (BGP) route only from its private WAN's Points of Presence (PoPs) closest to the cloud region. This ensures that user traffic transits Internet ISPs and enters the cloud provider's WAN at or near the cloud region's geographic location. In contrast, WAN-transit service advertises the BGP route from the cloud's PoPs globally.

The top-3 cloud providers (AWS, Azure and Google Cloud) offer such tiered services. Despite the dominance of cloud traffic in today's Internet ecosystem [5, 28], little is known about: (a) what routing strategies the cloud providers use for different service tiers; (b) how effective these strategies are in ensuring that the WAN-transit service offers better performance, given that BGP does not take path performance into account when selecting routes; and (c) whether better strategies exist to improve the performance of WAN-transit relative to inet-transit.

Previous work has explored some, but not all, of these questions, or has explored complementary questions (§8). Arnold et al. [25] compared Google's and AWS's WAN-transit and inet-transit services, but did not explore the potential latency improvement provided by alternative strategies or the specifics of the cloud providers' routing strategies. Moreover, their work was not able to examine one comparable giant cloud provider: Azure. Yeganeh et al. [90] investigated the latency and throughput of inter-cloud communications, not that of traffic from Internet users to clouds.

In this paper, we report results from a large-scale measurement study on tiered network services offered by the three largest cloud providers [6]: AWS, Azure, and Google Cloud. We use RIPE Atlas probes as vantage points to discover cloud edges where cloud traffic enters and exits a cloud provider and analyze the extent to which traffic stays within the cloud's WAN (§4). Furthermore, we study whether alternative routing strategies can offer better latencies than the cloud providers' existing services (§5). For this purpose, we develop a method to synthesize alternative routes using the cloud edges we uncover.

Our study uncovers several interesting and previously unknown findings about these services[2]. These findings hold in results from a repeated measurement campaign five months after the original measurements (§6). We find that:

(1) Cloud providers use a wide range of routing strategies (global anycast, regional anycast, and unicast) to implement their inet-transit and WAN-transit services (§4). Contrary to their advertised service descriptions [7, 23], some of them have overlapping routing strategies for their inet-transit and WAN-transit traffic, resulting in inet-transit traffic being carried on the cloud providers' own WANs.
(2) Although WAN-transit traffic often enters a cloud WAN early, it sometimes exits the WAN early so that return traffic to the user traverses the public Internet, contrary to cloud providers' goals for WAN-transit traffic [15, 23]. For instance, for Azure, we observe that more than 78% of our vantage points enter its WAN within a distance of 500 km, while only 61% of them exit the WAN within a distance of 500 km. About 12% of them exit the WAN more than 5000 km away from the cloud edge nearest the user (§4.4).
(3) The median RTTs and jitters of inet-transit and WAN-transit services of the three cloud providers we study are comparable, with WAN-transit being slightly better in some regions

[1] A cloud region denotes a geographic area containing VMs hosting one or more services.
[2] The data and code of this paper are accessible at https://github.com/SHiftLin/SIGMETRICS25-CloudRouting

and worse in others (§5.1 & §5.2). While this result is consistent with a previous study limited to two cloud providers [25], the overlapping routing strategies for inet-transit and WAN-transit services we uncover could partly explain the similarity in latency performance.

(4) We develop a method to explore and synthesize alternative routes that can potentially reduce latency. This enables us to quantify the potential for latency reductions. We find that a performance-based routing strategy can significantly reduce latencies in all three cloud providers for 4% to 85% of vantage point and cloud region pairs. For example, in one particular region of a single cloud provider there is the potential for over 100 ms latency reductions (§5.3) for a subset of vantage points, which is consistent with prior work [25].

Ethical considerations: This work does not raise ethical concerns. We used accrued and donated RIPE Atlas [82] measurement credits, alongside rented cloud virtual machines to send a limited amount of measurement traffic.

2 OVERVIEW

In this section, we describe the research questions that motivate this work and the challenges.

2.1 Goals

We aim to answer the following questions on cloud routing:

(1) Where does traffic from different service tiers (*i.e.*, WAN-transit and inet-transit) enter or leave a cloud's WAN? Specifically, how far away are traffic's ingress or egress edges from its closest cloud edges? From this, what can we infer about the routing strategies that cloud providers use for these service tiers?

(2) How well do current routing strategies perform in ensuring low latency? We are interested in latency because routing strategies directly impact end-to-end latency and it is an important metric for many Internet applications such as web, video conferencing, and gaming [37, 66]. Ideally, obtaining more metrics such as throughput can provide a more holistic view of cloud routing performance. However, we are limited by the constraints of the available measurement platform—RIPE Altas—to obtain throughput. Therefore, in this work, we focus on latency, specifically the median RTT.

(3) Are there alternative routing mechanisms that can improve the latency of the WAN-transit or inet-transit service?

Answering these questions helps us understand how well traffic from each service tier adheres to a cloud provider's intended transit policy. In addition, it helps us infer the routing strategies a cloud provider employs to offer WAN-transit or inet-transit services and understand how well these strategies work in providing low-latency routing.

2.2 Challenges

We are limited by a number of factors in answering these questions. First and foremost, we do not have ground truth for where traffic enters or leaves a cloud provider's WAN. A common approach to obtain such information is to use traceroute to detect AS borders and then infer the cloud WAN's ingress or egress IP address location [25, 26, 59, 63]. To achieve substantial coverage of cloud ingress locations, we should traceroute from many vantage points at diverse locations. We should also send many pings from these vantage points to measure the RTTs. However, the only large-scale research platform available for us to conduct such traceroute and ping measurements is RIPE Atlas [82], which limits both the amount and type of measurement traffic we can send. With this limitation, we must trade off between the number of RIPE Atlas probes we use and the amount of traffic each probe sends. The former affects how many distinct locations of clients are

represented, and the latter affects the statistical confidence of the metric we obtain. To address this challenge, we develop measurement techniques that can reduce the number of probes we use without significantly sacrificing coverage (§3.1) and the amount of measurement traffic while obtaining statistically confident results (§3.4).

While accurately geo-locating an IP address remains an open research challenge [38, 56], our study requires that we locate where a packet enters or leaves a cloud's WAN. To address this challenge, we build on the best current practice in this research area [25, 26, 39, 41, 60, 72] and develop a technique that geo-locates an IP address from multiple sources in the order of their perceived accuracy (§3.2).

Finally, we aim to explore whether alternative routing strategies can improve the latency to cloud services. However, we do not have access to the cloud routing systems and cannot change the existing cloud routes. For instance, we cannot change the egress cloud edges that cloud traffic takes by choosing different BGP next hops. Neither can we apply BGP control knobs such as AS prepending or Multi-Exit Discriminators (MEDs) to influence where traffic enters a cloud's WAN. To address this challenge, we develop a new measurement method to synthesize alternative routes that emulate different policy-compliant BGP next hop selections between a cloud provider and its peering ASes (§3.3) With this method, we are able to compare the current cloud's WAN-transit and inet-transit services with the alternative routing strategies.

3 MEASUREMENT METHODOLOGY

In this section, we describe the measurement infrastructures and elaborate on the methods we developed to solve the challenges identified in §2.2.

3.1 Measurement Infrastructures

We setup virtual machines (VMs) in multiple regions on three cloud providers, and use RIPE Atlas probes [82] as vantage points to launch network measurements to the VMs. RIPE Atlas provides ~12, 800 probes around the world at the time of our measurements, and each cloud provider provides more than 20 VM regions. We conducted two measurements consisting of several experiments in this study; the first was primarily in April/May 2024 and was repeated in Sep. 2024 (specific dates for each experiment in the measurements are in §3.5). We used the same method for both measurements and the results and conclusions are largely consistent. For ease of exposition, we present the results of the latest measurement and compare the two measurements in §6.

RIPE Atlas Probe Selection: To improve the stability, accuracy, representativeness, and efficiency of our measurements, we begin by selecting probes that have remained stable for at least 30 days, increasing the likelihood of consistent infrastructure throughout the study. Additionally, as our study uses a probe's geolocation information, we filter out the probes with potentially spurious geolocations, following the methodology in [41].

Furthermore, we obtain each probe's AS number (ASN) and the location—city—from the RIPE Atlas website, and group the probes by <ASN, city> as in previous work [25, 54, 64, 67, 92]. We then select one probe randomly from each group to conduct our measurements. We adopt this selection process because the probes in the same <ASN, city> group often enter a cloud's WAN via the same cloud edges. We validate this assumption by an experiment available in Appendix B. By selecting one probe from each <ASN, city> group, we save RIPE Atlas credits on measurements while maximizing cloud edge coverage. In total, we retain 5205 probes from distinct <ASN, city> groups for measurements. We show the geographic distribution of these probes in Appendix A. Overall, RIPE Atlas probes are disproportionately concentrated in North America and Europe and may be

Table 1. Selected regions' locations for each cloud. The two letters in parentheses are the abbreviations of regions.

Region	AWS	Azure	Google
Africa (AF)	Cape Town	Johannesburg	
Asia (AS)	Mumbai	Pune	Mumbai
Middle East (ME)	Tel Aviv	Doha	Tel Aviv
Oceania (OC)	Sydney		
Europe (EU)	Frankfurt		
North America (NA)	Virginia		
South America (SA)	São Paulo		

Table 2. Breakdown of each source used for IP to ASN mapping.

	Source	# (Percentage)
IP to ASN	BGP	31547 (77.3%)
	IP Ranges	6226 (15.3%)
	CAIDA IXP	733 (1.8%)
	Unidentified	2290 (5.6%)
	Total	40796 (100.0%)

underrepresented in other regions of the world [64, 92]. As a result, the latency measurements in this study primarily reflect the experiences of clients in North America and Europe regions.

Cloud Providers: We study the three largest providers: AWS, Azure, and Google Cloud. All have private backbones and offer two classes of networking services: WAN-transit and inet-transit. Unlike the others, AWS offers its WAN-transit service at the transport layer. When AWS's WAN-transit traffic enters its private WAN, a proxy (aka Global Accelerator [7]) intercepts a TCP connection or a UDP packet and then dispatches it to a VM. In contrast, Azure and Google Cloud offer both tiers of services at the network layer. In all three providers, the IP address of a VM identifies the service tier.

Cloud Region Selection: For each cloud, we select seven regions to setup the VMs as shown in Table 1. Conducting experiments on all the cloud regions of all three cloud providers is too costly both in terms of measurement traffic and finance, so we select regions from across the globe. We also believe that selecting disparate cloud regions potentially reflects accurate conditions for users as data sovereignty concerns have become more prevalent and large network services have recently run afoul of legislation [34, 57], so users may be reaching back to remote regions to access their data more frequently in the future.

We use these regions because they cover all continents except Antarctica and are located in the major population and metropolitan areas of those continents. We aim to choose overlapping cloud regions to enable cross-provider examination whenever possible. However, some cloud providers do not have overlapping open-access regions in Africa, Asia, and the Middle East, so we choose their nearby cloud regions instead.

We setup one VM in each region. With each VM, we bound two IP addresses: one is routed by the inet-transit service and the other by the WAN-transit service. By launching network measurements from probes to the corresponding IP addresses, we can observe the performance of different transit services in different regions of a cloud provider. All VMs run Ubuntu 22.04. We configure each VM with 4 CPUs and 8 GiB memory, which ensures that the CPU and memory do not become a bottleneck in the experiments; our monitoring logs show that all VMs' CPU utilization and RAM utilization are below 5% and 10% during our experiments, respectively.

3.2 Cloud Edge Discovery

Our first measurement goal is to uncover where traffic enters or leaves a cloud provider's network. We accomplish this goal by detecting the ingress or egress edges of the cloud provider's network along the path taken. This problem is a special and simpler instance of the problem addressed by bdrmap [59] and bdrmapit [63], which detect the borders between any two ASes in a traceroute

output. Therefore, we adopt and simplify the approaches in [25, 26, 77, 92] to detect cloud edges. We briefly summarize our approaches for purposes of reproducibility and completeness.

Specifically, to identify where traffic leaves or enters a cloud, we instrument all selected probes to launch traceroutes to all seven VMs of the cloud. Conversely, we also traceroute out from VMs to selected probes. We then annotate each public IP address in a traceroute output with its ASN and geolocation information as follows.

For IP to ASN mapping, we gather ground truths from three sources: BGP routing data collected from the RouteViews servers [22, 44], cloud provider self-published IP ranges [8, 10, 17], and CAIDA's IXP dataset [32], which is a superset of PeeringDB [21], Hurricane Electric [11], and Packet Clearing House [19]. Table 2 shows the percentage of IP addresses whose ASNs are successfully identified by each source in the order of precedence from top to bottom.

We geolocate an IP address using five sources of information, in the decreasing order of their perceived accuracy as described below. That is, we consult a source of lower accuracy only if we cannot geo-locate an IP address with sources of higher accuracy.

Geofeed: We collect IP Geolocation feeds (Geofeeds) [31, 52] from multiple sources, including OpenGeoFeed [18], WHOIS [33, 36], and RIPE Database [69]. Geofeeds are self-published datasets of IP-to-geolocation mapping by network operators such as ISPs, cloud providers, and CDN providers. Since the locations are provided by the owners of the IP prefixes, we consider this method have the highest degree of accuracy.

IXP Dataset: Existing IXP datasets provide not only the IP addresses that an IXP assigns to its connected ASNs but also the geolocations of IXP facilities. Therefore, we can use these datasets to geolocate those IP addresses. If an IP address belongs to an IXP facility's network, we assign the geolocation of the facility to the IP address. Previous work shows that 99% of parsable PeeringDB records are correct [80]. Therefore, we use PeeringDB and consider it highly accurate.

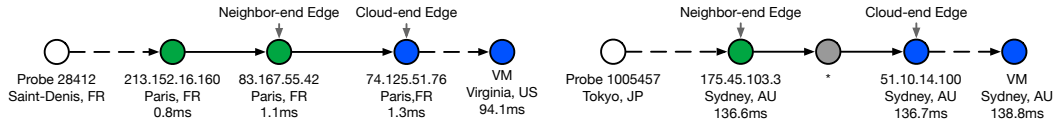
rDNS: We use `hoiho` [60], a reverse DNS (rDNS) based IP geolocation system as the third source for geolocation. Prior work shows that `hoiho` has an accuracy of 94% [60].

RIPE IPmap: RIPE IPmap is a proactive method to infer the geolocation of an IP address by measuring the RTTs from multiple RIPE Atlas probes to the IP address [39]. Existing research [39] shows that RIPE IPmap is highly accurate at the granularity of 500 km (Figure 1 in [39]). Therefore, we only report performance numbers at this or coarser distance granularity.

Geo-IP Database: If we cannot geo-locate an IP address by the four sources above, we resort to a commercial Geo-IP database IPinfo [12] as our last source of truth for IP geolocation. IPinfo also uses a probe-based IP geolocation method with more than 600 probe servers around the world [84] and we find it to be more accurate than MaxMind [4] and IP2Geolocation [3]. We cross-examined the geolocation results returned by IPinfo and RIPE IPmap on the subset of edge IPs (1452) that are geolocatable by both services. We find that for 90% of them, the distance difference is less than 100 km; and for 95% of them, the distance difference is less than 500 km.

In addition to the above IP geolocation sources, we also apply the published cloud edge locations [9, 14, 20] and PeeringDB [21] to filter potential errors. If a cloud's edge IP address in a traceroute output is resolved to a location where a cloud does not have a published presence within 50 km, we discard the traceroute. Moreover, we use the RTTs from traceroute packets to filter any hops whose IP geolocation results violate the speed-of-light constraint.

We use the following method to identify a cloud edge as illustrated by two examples in Fig. 1. In an inbound traceroute from a probe to a VM, we detect a cloud edge IP as the first occurrence of an IP address that belongs to a cloud provider's address space. We call this IP as "*cloud-end edge IP*", and its preceding responsive IP as "*neighbor-end edge IP*". For a valid cloud edge detection, we require that either there are no unresponsive hops between the cloud-end edge IP and the



(a) Edges *not* separated by unresponsive hops (b) Edges separated by unresponsive hops (*)

Fig. 1. This figure shows how we detect cloud edges from traceroutes. The green circles represent the hops in ISPs, and the blue ones are the hops in clouds. A gray circle indicates an unresponsive hop. A valid cloud edge requires either (a) the neighbor-end and cloud-end edge IPs are *not* separated by unresponsive hops, or (b) they are separated by unresponsive hops, but their geolocations are resolved to the same city (distance < 50 km) and their RTT difference is less than 2 ms.

Table 3. The number of cloud edge metros, cloud-end, and neighbor-end edge IPs discovered. Some neighbor-end IPs overlap among cloud providers.

	# Edge metros		
	Discovered	Published	%
AWS	82	97	84
Azure	99	105	94
Google	75	107	70
	# Edge IPs		
	Cloud-end	Neighbor-end	
AWS	1431	1993	
Azure	1137	2503	
Google	1130	2455	

Table 4. Breakdown of each source used by geolocating both cloud-end and neighbor-end edge IPs

	Source	**# (Percentage)**
Edge IP Geolocation	Geofeed	261 (2.6%)
	PeeringDB	576 (5.8%)
	rDNS	325 (3.3%)
	RIPE IPmap	3105 (31.1%)
	IPinfo	5702 (57.2%)
	Unidentified	0 (0.0%)
	Total	9969 (100.0%)

neighbor-end edge IP (Fig. 1a), or the two IP addresses are located in the same metropolitan area. To meet the latter condition, we require that the neighbor-end edge IP and the cloud-end edge IP's geolocations are resolved to the same city (< 50 km) and the RTT between the two IPs is less than 2 ms. (Fig. 1b). In this latter case, we consider the detected cloud edge to be sufficiently close to the true cloud edge even if some unresponsive hops belong to a cloud. Ditto for the outbound cloud edge detection.

In our traceroute data, we also observe Multiprotocol Label Switching (MPLS) [76] labels embedded in ICMP extensions [29], indicating tunnel existence in the path. MPLS tunnels do not affect our border detection method. Currently, cloud WANs and ASes commonly use MPLS internally, but MPLS does not cross the AS boundaries. Thus, when the cloud WAN and the neighbor AS both use MPLS, the cloud-end and neighbor-end edges will be the endpoints of their respective MPLS tunnels. Since one tunnel ends and another begins as the boundaries are crossed, both edges will reply to our traceroute packets. Thus, when MPLS tunnels exist, our method still correctly detects the cloud edges.

Once we detect a cloud edge, we record the edge IP address and geolocation as well as the preceding IP address and its ASN. If the above cloud edge detection procedure is successful, then for each traceroute output between a ⟨probe, VM⟩ pair, we obtain its ingress/egress cloud edge location and ingress/egress AS information.

Table 3 summarizes the number of cloud edges we uncover from traceroutes by metro locations and IP addresses. We also list the number of cloud edges published by each cloud provider on its

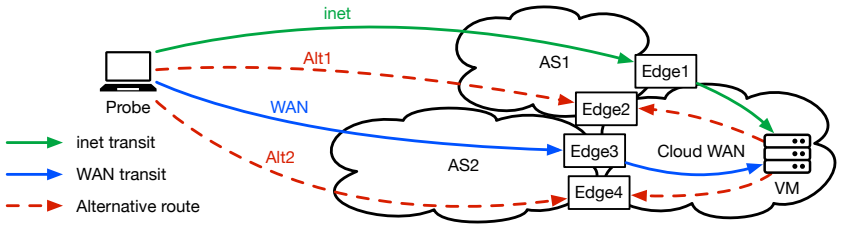


Fig. 2. We synthesize alternative routes by choosing different cloud edges for traffic to ingress or egress a cloud's WAN. Edge1 and Edge3 are the original ingress edges the inet-transit and WAN-transit traffic traverse. Their ASes also peer with the cloud's WAN at Edge2 and Edge4. We synthesize two alternative routes (Alt 1 and Alt 2) and send ping packets from a probe to Edge2 (or Edge4) and from a VM to Edge2 (or Edge4) to measure their latencies.

website. We uncover 84% and 94% of AWS and Azure's reported edges, respectively. For Google, we uncover 70% of its reported edges. As we do not aim to uncover all cloud edges, we do not run traceroutes from VMs to the entire IP address space, hence the limited coverage. The relatively low coverage for Google is possibly due to how it announces inet-transit prefixes as we will show in §4.1. Table 4 shows the distinct cloud edge IP addresses geolocated by each source in our traceroute outputs. We discard the IP geolocation results with speed-of-light violations.

3.3 Exploring Alternative Routes

Another of our measurement goals is to explore whether alternative routing mechanisms can improve cloud transit services. To meet this goal, we must measure the latencies of alternative routes. As BGP [74] determines the inter-domain routes, a cloud provider can explore the performance of alternative routes by adjusting how it announces an IP prefix to its BGP peers [54, 64] or how it selects BGP next hops [79, 88]. However, unlike previous work [54, 64], we do not have access to a cloud provider's routing system, and therefore cannot modify BGP advertisements to create alternative routes. Besides being inaccessible to most researchers, such an approach is time-consuming, because it usually requires iterating during multiple rounds of different announcement strategies and has to wait for BGP convergence before evaluating the effectiveness in each round.

To address this challenge, we emulate alternative routing strategies by synthesizing alternative policy-compliant routes[3]. We decompose an end-to-end path between a probe and a VM into two segments: one segment between the probe and a cloud edge and the other segment between the cloud edge and the VM. We synthesize an alternative route by choosing an alternative cloud edge different from the original ingress (or egress) edge between the probe and the VM. Recall that after the cloud edge discovery step (§3.2), We obtain a "*cloud-end edge IP*" and a "*neighbor-end edge IP*" for each detected cloud edge. We consider an AS as an "*ingress AS*" if any of its IP addresses appears in a traceroute output as a neighbor-end edge IP. For each ingress AS, we collect all cloud edges (IPs and metros) that peer with the ingress AS, as seen from the traceroute outputs. If in one traceroute output, we observe that a probe reaches a cloud edge IP via an ingress AS, we assume that all cloud edges that peer with the same ingress AS are policy-compliant edges for carrying this probe's traffic. This is because common BGP policies are specified at the AS level [24, 40]. This method for synthesizing an alternative route emulates the scenario where a cloud provider (or its peer) selects a different BGP next hop to reach a probe (or a VM).

[3]In this paper, we use "route" to refer to a virtual control plane construct that stitches together the next hop entry to an IP prefix in each router's forwarding table; we use "path" to refer to the data plane path a packet takes.

Fig. 2 shows an example. From the traceroute outputs, we discover two ingress ASes between a ⟨probe, VM⟩ pair. Each AS peers with the cloud WAN at two cloud edge locations. Edge 1 and Edge 3 are discovered by the original inet-transit and WAN-transit traceroute from the probe to the VM, and we discover Edge 2 and Edge 4 through the traffic from other probes to the cloud. Then when we synthesize the alternative routes for the ⟨probe, VM⟩ pair in the figure, we can choose Edge 2 or Edge 4 as the alternative cloud edge. For instance, in the figure, from Probe to Edge 2 and then to VM is an alternative route.

Once we synthesize an alternative route, we measure its RTT by sending pings from a probe and a VM to the alternative cloud edge. At each cloud edge, there exist multiple cloud-end edge IPs that peer with different ASNs. Not all of them respond to ping packets, or a cloud-end edge IP may not be routable as a destination since it may not be announced in BGP. In these cases, we also attempt to use the neighbor-end edge IPs that belong to a cloud's peer ASes and respond to pings as alternative cloud edges.

Since we can only measure RTTs instead of one-way latency between a probe (or a VM) and a cloud edge, we synthesize only symmetric alternative routes that share the same ingress/egress cloud edges. As shown in Fig. 2, Edge2 (or Edge4) is both the ingress and egress cloud edge of an alternative route. We add the RTT from a probe to a cloud edge and the RTT from a VM to the cloud edge to approximate the full RTT of an alternative route. Today's destination-based Internet routing may not take such a synthesized route via an edge candidate [48, 51]. However, such routes would have been viable had the routing system chosen them. In this paper, we aim to explore these routes' latency improvement if ISPs and cloud providers adjust their routing policies to route packets through these synthesized paths.

We use the median RTT as a performance metric when evaluating the performance of a path, as it is robust against outliers [35, 37, 55, 91, 92]. We also compute latency jitter [55] as deviation around the median for latency measurements for WAN-transit and inet-transit traffic.

In our measurements, we predominantly use ICMP pings whenever possible and use DNS pings when ICMP pings are unavailable. For instance, we cannot use ICMP pings to measure the RTTs of AWS Global Accelerator [7], as it is a proxy service that intercepts application-layer traffic. We compared the latencies measured by DNS pings and ICMP pings and did not observe significant differences. Therefore, unless specified otherwise, we do not differentiate between ICMP and DNS ping results.

We may not observe all ASes peering with a cloud provider from our traceroutes. Therefore, the candidate alternative cloud edges we use are likely to be only a subset of all policy-compliant cloud edges a probe can use to enter or leave a cloud. Thus, this study serves as a lower-bound estimate of how many alternative routes exist and how much they can improve the latency of cloud routing.

3.4 Balancing Efficiency and Confidence

Obtaining the RTT estimate of a path with statistical confidence requires considerable RTT measurements. However, to explore the performance of an alternative path, we need to obtain the RTTs from a probe to all of its alternative cloud edges, which would require a prohibitively large number of pings to obtain the RTTs of all alternative paths.

To address this problem, we trade off coverage for efficiency. First, we pre-screen alternative routes by sending a few pings—ranging from three to nine—to all alternative routes. Only if the minimum RTT of an alternative path is less than that of a WAN-transit path, we consider it as a potentially better alternative path. Moreover, a ⟨probe, VM⟩ pair may include multiple alternative paths, and we selected only one alternative path whose minimum RTT is also the minimum among those alternative paths, because such a path has the greatest potential for improving the latency of cloud routing. We then launch repeated ping measurements along those pre-screened alternative

Table 5. The dates of all experiments in this work. The TraceClouds experiment for Azure is on 2024/01/25, which is earlier than the others. This will not affect our results and conclusions because TraceCloud is only used for cloud edge discovery and is *never* used for any RTT estimate.

	First Measurement			Second Measurement		
	TraceClouds	PingEdges	RepeatPings	TraceClouds	PingEdges	RepeatPings
AWS	2024/04/20	2024/04/27	2024/05/11	2024/09/24	2024/09/24	2024/09/26
Azure	2024/01/25	2024/03/26	2024/05/10	2024/09/10	2024/09/10	2024/09/12
Google	2024/04/07	2024/04/13	2024/05/09	2024/09/03	2024/09/03	2024/09/05

paths to obtain median RTTs. Concurrently, we also send repeated pings from probes to VMs by both inet-transit and WAN-transit services so that we can compare the median RTTs between alternative paths and original paths. For each RTT measurement, we send 96 pings over 24 hours and use the median of the 96 RTT samples as the measured median RTT value.

The selected alternative path(s) in the pre-screening step are not required to consistently have the minimum RTT. There may exist other paths with lower RTTs, or the paths we selected may not be the minimum in the future. However, by running repeated pings on these selected alternative paths for 24 hours, we investigate whether they can provide consistent latency improvement (§5.3). It is possible that some alternative paths not selected for the repeated ping measurements may consistently provide more improvement. This omission likely underestimates the number of better alternative routes, making the improvements we observe a lower bound to potential improvements of alternative routing strategies.

3.5 Temporal Validation

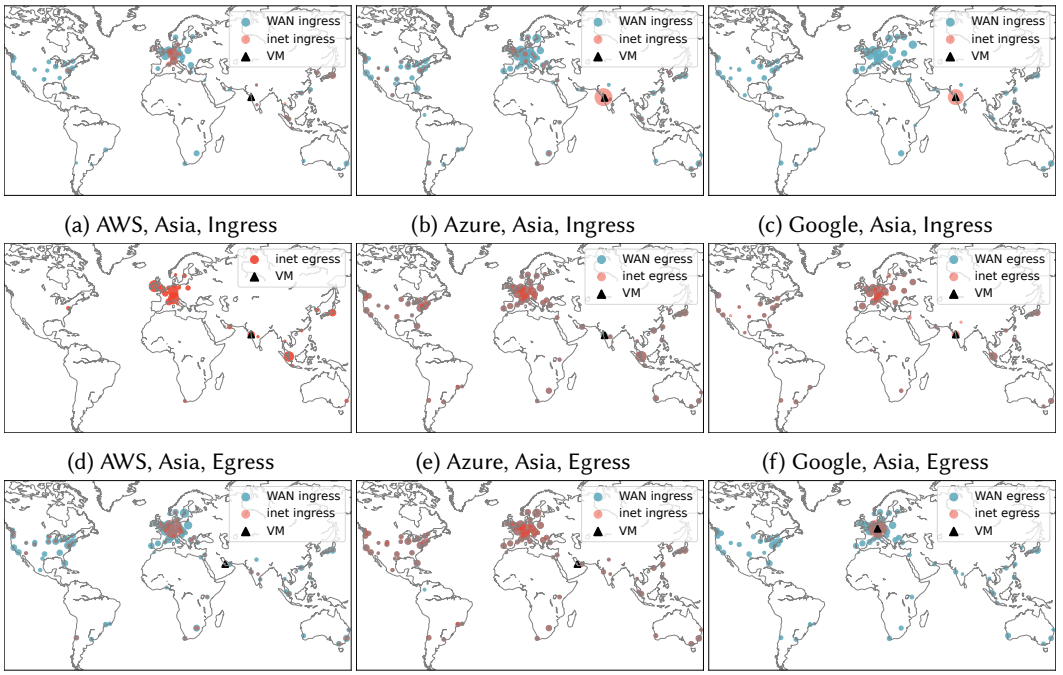
As presented in previous sections, our measurement consists of three experiments. Firstly, we instrument RIPE Atlas probes to send pings and traceroutes to the cloud VMs through different tiers of network services and vice versa. We refer to this experiment as "`TraceClouds`". With this experiment's data, we obtain the cloud edges (§3.2) and synthesize the alternative routes (§3.3). Then we use both probes and VMs to ping all discovered edges to obtain alternative routes' preliminary RTTs (§3.4). We refer to this experiment as "`PingEdges`". Finally, with the preliminary results of these alternative paths' RTTs, we select the routes with potential improvement and launch repeated pings to the selected routes (§3.4). We refer to this experiment as "`RepeatPings`".

We ran the measurement in the early stage of our study, and after around five months, we ran the measurement again to observe whether our conclusion persists. We updated all relevant data sources used for IP-to-ASN mapping (§3.2) and IP geolocation (§3.2) during the analysis of our second measurement. We list the start date of all experiments in these two measurements in Table 5. Every experiment was finished within 24 hours. As we will present in §6, all conclusions presented in our paper are persistent and verified even after five months. Unless otherwise specified, we present the results of the most recent measurement in this paper.

We selected 5831 probes with unique <ASN, city> attributes in the first measurement, but some probes are disconnected after a few months with 4159 selected probes remain connected. We then randomly select newly connected probes in those missing <ASN, city> locations. Thus, in the second measurement, we select 5205 probes representing 5205 <ASN, city> groups (§3.1).

4 INFERRING ROUTING STRATEGIES

In this section, we describe our findings on where traffic enters or exits each cloud provider's WAN. These findings enable us to infer how cloud providers announce the WAN-transit or inet-transit



(a) AWS, Asia, Ingress (b) Azure, Asia, Ingress (c) Google, Asia, Ingress

(d) AWS, Asia, Egress (e) Azure, Asia, Egress (f) Google, Asia, Egress

(g) Azure, MiddleEast, Ingress, Jan. (h) Azure, MiddleEast, Ingress, Sep. (i) Azure, Europe, Egress

Fig. 3. Geographical distributions of the ingress/egress cloud edges for RIPE Atlas probes to reach different network service tiers of the Asia region for three cloud providers (a-f) and the Middle East and Europe regions for Azure (g-i). The size of a circle indicates the number of probes ingressing/egressing the cloud edge. The distinct patterns reflect the diverse routing strategies employed by the cloud providers. Egress edges of AWS' WAN-transit service are unavailable.

IP prefixes to offer tiered services. This knowledge in turn will help us understand the impacts of routing mechanisms on end-to-end latencies.

4.1 Ingress Cloud Edges

We use the distribution of ingress cloud edges from probes to VMs to infer a cloud provider's WAN-transit and inet-transit services' routing strategies. In order for traffic destined to a VM to enter a cloud edge at a metro location from a peer AS, in BGP a cloud provider must announce the VM's IP prefix to the peer at that metro. Thus, if we observe a cloud edge in an ingress traceroute from a probe to a cloud VM, we can infer that the cloud provider has announced the VM's IP prefix at that cloud edge.

We observe different routing strategies by different cloud providers for providing WAN-transit and inet-transit services. For each strategy we observe, we illustrate it with a representative example in Fig. 3. Besides, we use the entropy metric to quantify the variety of the ingress/egress edge distributions of a cloud's network services. Entropy indirectly estimates the diversity of cloud routing strategies.

Specifically, we estimate the probability (p_e) that a pair of ⟨probe, VM⟩ ingresses (or egresses) at a cloud edge as the number of ⟨probe, VM⟩ pairs ingressing (or egressing) at the cloud edge e divided by the total number of ⟨probe, VM⟩ pairs. We compute the entropy metric as $-\sum_e p_e \log_2 p_e$,

ranging from 0 to $\log_2 N_e$, where N_e is the total number of observed cloud edges. A larger value of entropy indicates more evenly distributed cloud edges. As shown in Table 3, the numbers of cloud edges (by metro locations) for the three clouds range from 97 to 107. Therefore, the maximum entropy is around $log_2 107 \approx 6.7$.

Global Anycast for Ingress WAN-transit Traffic: Fig. 3 (a, b, c) show three representative examples of the ingress edge distributions in our data. It plots on a world map the ingress edge distributions for the south Asia region of three clouds. The blue circles show the locations of the ingress edges of their WAN-transit traffic, with the size of each circle indicating the number of probes that enter the cloud at that location. We observe that the blue circles are globally distributed without any large dominant ones, indicating that the ingress WAN-transit traffic is spread out over these cloud edges. The entropy values of these three distributions are larger than 4.9. From this result, we infer that all three clouds announce a WAN-transit IP prefix globally across their cloud edges. This strategy is consistent with the description of the WAN-transit service on each provider's website [7, 15, 23].

However, we observe significant differences in how the cloud providers offer the inet-transit service. We plot five distinct patterns using the red circles in Fig. 3 (a, b, c, g, h).

Regional Anycast for inet-transit Traffic: For AWS, we observe that the inet-transit traffic ingresses from various edges in the continents (Asia and Europe) surrounding the VM's cloud region and at one U.S. east coast location, as shown by the red circles in Fig. 3a. In addition, there is no dominant ingress cloud edge that attracts the majority of the traffic. The entropy values of the distributions for different AWS regions range from 1.9 to 3.8. Therefore, we infer that AWS adopts a routing strategy that is analogous to regional anycast [45, 61, 92] and announces an inet-transit IP prefix from the cloud edges in nearby geographic areas.

Unicast for inet-transit Traffic: In contrast, for Google, all inet-transit traffic enters Google at the cloud region where the VM is located, as indicated by the single large red circle in Fig. 3c. We also find its entropy value is 0. We infer that Google announces an inet-transit VM's IP prefix only at the VM's cloud region. This strategy enables the inet-transit traffic to transit over the public Internet to reach the VM, consistent with the service description on Google's website [15]. This could also explain why we observe fewer Google Cloud edges than AWS and Azure (Table 3), as it does not announce the inet-transit prefixes as broadly as the other clouds.

Azure's ingress edge distribution for its inet-transit traffic (Fig. 3 (b, g, h)) is the most puzzling among the three, and we observe three distinct patterns.

Predominantly Unicast Locally for inet-transit Traffic: Different from Google, Azure's ingress edges are globally distributed, although the VM is located in Asia. Different from AWS, although the ingress edges are globally distributed, the majority of the probes (95.5%) enter the cloud in the cloud edge near the VM, as indicated by the dominant large red circle at the VM location in Fig. 3b. This distribution corresponds to an entropy value of 0.5. From this observation, we infer that Azure announces an inet-transit IP prefix at the VM region as well as selectively announcing it to a subset of peers at globally distributed cloud edges. This setup is perhaps due to traffic engineering efforts.

Predominantly Unicast in a Different Region for inet-transit traffic: As shown in Fig. 3g, in Jan. 2024, we observe that for Azure's VM located in Doha, Qatar, the majority (96.2%) of the probes reach the VM via a cloud edge in Frankfurt, Germany. As 65.9% of RIPE Atlas probes we use are located in Europe, there is still a significant percentage of non-European probes enter Azure's WAN in Frankfurt. Besides, we find that this distribution's entropy is 0.4. From this ingress edge distribution, we infer that Azure predominantly announces an inet-transit prefix in a region different from a VM's region. This ingress policy again makes Azure become the transit provider for the inet-transit traffic from Frankfurt to Doha, inconsistent with its website description [23].

Table 6. Inferred routing strategies cloud providers use to announce inet-transit and WAN-transit IP prefixes.

	WAN-transit	inet-transit
AWS	Global Anycast (Fig. 3a)	Regional Anycast (Fig. 3a)
Azure	Global Anycast (Fig. 3b)	Predominantly Unicast Locally (Fig. 3b)
		Predominantly Unicast in a Different Region (Fig. 3g)
		Global Anycast (Fig. 3h)
Google	Global Anycast (Fig. 3c)	Unicast Locally (Fig. 3c)

Global Anycast for inet-transit Traffic In the second measurement conducted in Sep. 2024, we observe globally distributed ingress edges for the inet-transit VM located in Doha for Azure, as shown in Fig. 3h. The edge distribution for inet-transit traffic overlaps with that for WAN-transit traffic and achieves an entropy value of around 5.3. Besides the Qatar region, we observe such global anycast for the inet-transit IP prefix originated from Azure in Africa for both sets of measurements. Therefore, we infer that Azure announces some inet-transit IP prefixes globally in the same manner as WAN-transit prefixes. This strategy is again inconsistent with the service description on Azure's website [23].

In summary, we observe ingress policies of AWS and Azure that are not consistent with those on their websites [7, 23]. When AWS or Azure announces an inet-transit VM's IP prefix at a cloud edge different than the VM region, the ingress traffic traverses the provider's WAN from the cloud edge to the VM, not over the Internet.

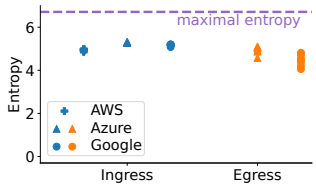
For AWS and Google respectively, we observe that they adopt the same ingress routing strategies for the inet-transit traffic for seven VM regions, although their routing strategies are different from each other. However, for Azure, we observe three different types of ingress edge distributions for the inet-transit traffic in seven VM regions, hence three different routing strategies. Table 6 summarizes the inferred ingress routing strategies of the three clouds.

In Fig. 3b, the few probes that ingress Azure from cloud edges other than the VM's region raise one question whether the result is due to IP geolocation errors that mistake a cloud edge IP located in the VM's region to a different location. To answer this question, we manually examine ten ingress edge IPs using looking-glass servers. For ten out of ten cases, we observe the presence of the edge IPs at the looking-glass locations (see Appendix C for more detail). In addition, those ingress edges are far away from the cloud region in Asia and such distance errors are detectable by active-probing-based geolocation methods used by IPmap [39] and IPinfo [12]. We did not extend this manual validation to more ingress edge IPs, because the process is manual and time-consuming, and we have gained confidence from the small validation set that the observed ingress edges are not due to IP geolocation errors.

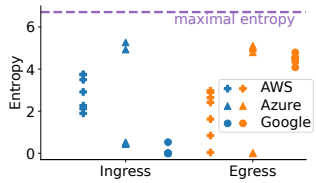
4.2 Egress Cloud Edges

We obtain the probe traffic's egress cloud edges from the traceroute outputs in the VM-to-probe direction. We use `scamper` [58] to run the traceroutes. We are unable to obtain the egress cloud edges for AWS's WAN-transit traffic because it is a proxy-based service [7], which does not allow users to initiate a connection inside the cloud to the Internet by the WAN-transit service. Hence, in this work, we only show the egress results of its inet-transit traffic.

Similar to ingress, we observe different egress edge distributions for different providers and within the same provider. We show a few representative examples in Fig. 3d, 3e, 3f, and 3i. Fig. 3d–3f shows the egress edge distributions for the three clouds in the south Asia region. Azure and Google have similar egress edge distributions for their inet-transit and WAN-transit traffic in that region,



(a) Entropy (WAN-transit)



(b) Entropy (inet-transit)

	WAN	inet
AWS	N/A	43%
Azure	54%	62%
Google	64%	15%

(c) Symmetry

Fig. 4. (a) and (b): The entropy metric that captures the ingress/egress edge diversity of ⟨probe, VM⟩ pairs. (c) Percentage of ⟨probe, VM⟩ pairs that use the same ingress and egress cloud edges. Overall, this figure summarizes the clouds' diverse routing strategies for their WAN-transit and inet-transit service.

as shown by the overlapping red and blue circles in Fig. 3e and 3f. Their distributions' entropy values are large than 4.8. AWS, on the other hand, egresses its inet-transit traffic at a subset of its cloud edges near the VM region, as shown by the red circles in Fig. 3d. The corresponding entropy values range from 0.04 and 3.0 for different regions in AWS. From these observations, we infer that Azure and Google have similar routing strategies for their egress traffic, while AWS uses a different exit policy.

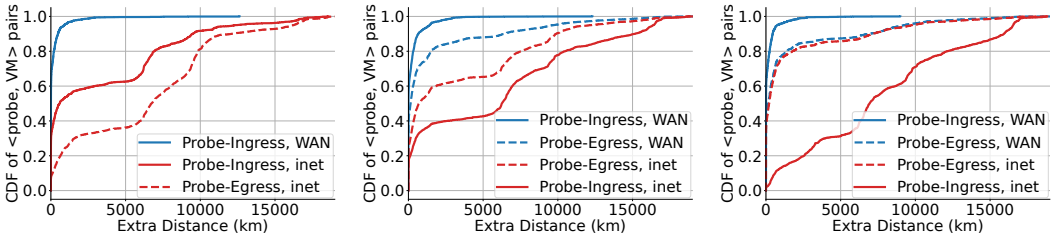
For Azure, we actually observe different egress edge distributions, hence different exit strategies. While in Fig. 3e, both inet-transit and WAN-transit traffic exits the cloud's WAN at globally distributed edges, in Fig. 3i, Azure's inet-transit traffic exits the cloud's network near the VM region, and it shows a near-zero entropy value. The edge distribution in Fig. 3i is consistent with Azure's service description [23].

Why does a cloud provider deliver its inet-transit traffic across its backbone as observed in Fig. 3e and 3f? One of the cloud providers has an explanation on its website [16]. The cloud provider early-exits inet-transit traffic near a VM region only when the traffic volume reaches a threshold. This policy can affect measurement studies that compare the performances of a cloud provider's inet-transit and WAN-transit services [25], because if the volume of the measurement traffic is low, both inet-transit and WAN-transit traffic may share the same egress paths. If we increase the levels of inet-transit traffic volume, it is possible that the egress edges of the inet-transit traffic would change. We did not attempt to validate this hypothesis due to the scale of this work and the cost to send a large amount of traffic. We defer the study of egress changes to future work.

4.3 Edge Diversity and Symmetry

To summarize the diversity of edge distributions of different network services for three cloud providers, we plot the entropy values in Fig. 4a and Fig. 4b. We use color to encode the ingress or egress direction, and shapes are used to encode different clouds. Each marker represents an data point from one of the regions. As indicated by the blue clusters at the top of Fig. 4a, all three clouds have high entropy values for their WAN-transit ingress traffic, as they announce the WAN-transit services' IP prefixes globally across their WANs. Azure adopts different routing strategies for its inet-transit service, as indicated by the wide ranges of its ingress and egress entropy values (the blue and orange triangles in Fig. 4b). AWS announces its inet-transit traffic more broadly than Google, as shown by the blue crosses and circles. Finally, the similarity between Google's egress strategies for its WAN-transit and inet-transit traffic can be observed from the similar entropy values of its egress edge distributions for these two services (the orange circles in Fig. 4a and Fig. 4b).

We also calculate the percentages of ⟨probe, VM⟩ pairs ingressing and egressing a cloud at the same metro location and show the results in Fig. 4c. For AWS and Google, the majority of



(a) AWS (b) Azure (c) Google

Fig. 5. The cumulative distributions of the extra distance a probe travels compared to entering or leaving a cloud via its closest cloud edge for both WAN-transit and inet-transit services. We cannot obtain the egress edges for AWS's WAN-transit traffic.

inet-transit traffic has asymmetric ingress and egress edges, but Azure route 62% of inet-transit traffic symmetrically. For the WAN-transit traffic, Google has the highest symmetry, with 64% of the traffic ingressing or egressing via the same metro location.

4.4 Extra Distances

Given the high degree of asymmetry between ingress and egress edges, we ask the question: how often does the WAN-transit service enter or leave a cloud's WAN near its closest cloud edges? The answer to the question helps us understand how clouds serve customers' traffic in their WANs. For this purpose, we compute two geographical distances: one is the distance between a probe to its closest cloud edge, and the other is the distance between a probe and its actual (ingress or egress) edge to a cloud VM. We then plot the difference between these two distances and show the cumulative distribution of the differences for all ⟨probe, VM⟩ pairs in Fig. 5.

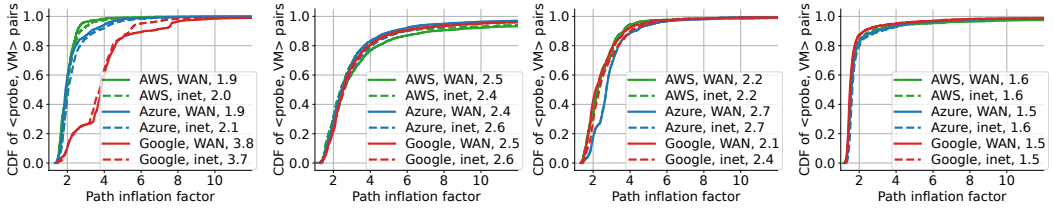
From Fig. 5, we observe that 59% (AWS), 55% (Azure), and 59% (Google) of the WAN-transit ⟨probe, VM⟩ pairs enter a cloud's WAN at the probes' nearest cloud edges, as indicated by the y-intercepts of the solid blue lines in the figures. However, only 38% (Azure) and 38% (Google) of the WAN-transit ⟨probe, VM⟩ pairs exit a cloud's WAN at the probes' closest cloud edges. Furthermore, for about 12% (Azure) and 13% (Google) of the WAN-transit ⟨probe, VM⟩ pairs, the extra distance that a probe's traffic travels from its egress edge exceeds 5000 km.

For completeness, we show both the CDFs of the distance from a probe (or a VM) to the ingress/egress cloud edge for all ⟨probe, VM⟩ pairs in Fig. 10 and Fig. 11 in Appendix D. We find that for WAN-transit service, about 84% (AWS), 78% (Azure), and 81% (Google) of probes enter a cloud's WAN within a distance of 500 km, but only about 61% (Azure) and 61% (Google) of ⟨probe, VM⟩ pairs' traffic exit the WAN within a distance of 500 km to the probe. Overall, the trend in these distributions is similar to that shown in Fig. 5: WAN-transit traffic enters a cloud's WAN early, but a significant fraction of it also leaves the cloud WAN early.

The above results show that Azure and Google do not always route their WAN-transit traffic to the cloud edges nearest to the destinations, despite the fact they have complete control over their egress traffic [49, 88]. Such a routing strategy may come from different goals beyond exclusively latency when the cloud providers route customers' traffic, such as achieveing a lower packet loss or a higher bandwidth. We leave the investigation of their concrete purposes for future work.

5 IMPACTS ON LATENCY

From the previous section, we observe that cloud providers employ different routing strategies for providing different tiers of network services. Since routing has direct impacts on end-to-end



(a) Asia (b) Europe (c) Middle East (d) North America

Fig. 6. CDFs of the *path inflation factor* for ⟨probe, VM⟩ pairs. The numbers in the legend indicate the median values of the *path inflation factor* of each service.

latencies, we study these impacts in this section. We also introduce alternative routing strategies and their potential impacts on end-to-end latencies.

5.1 Path Inflation Factor

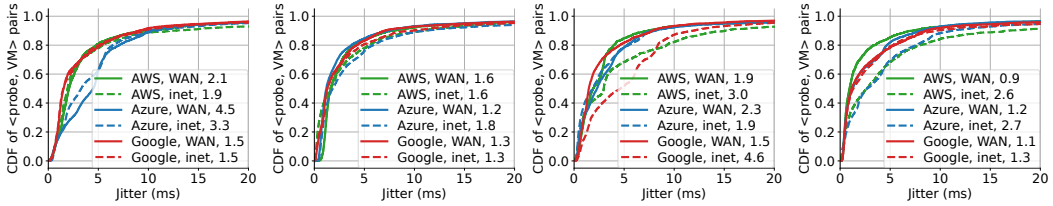
We measure the RTTs between ⟨probe, VM⟩ pairs for different clouds and for different tiers of services. Since comparing RTTs across different cloud providers is challenging, as their cloud regions are not exactly colocated, we use a distance-normalized latency metric: the *path inflation factor* [30], to quantify how "fast" a route is at moving the traffic between two endpoints. Specifically, to compute the path inflation factor, we first compute the speed of traffic v by dividing the distance between a ⟨probe, VM⟩ pair by the RTT between the pair. We then normalize this speed by the speed of light in fiber c: c/v. The larger this number, the slower the traffic travels. We consider a route as more efficient if it leads to lower path inflation. We use the measured median RTT values of the repeated experiments mentioned in §3.4 to compute the path inflation factors.

Fig. 6 shows the path inflation factors for the different clouds and network service tiers. Due to space limitations, we show four representative regions for each cloud and present other regions' results in Appendix E. Firstly, we observe that the path inflation factors of a cloud's WAN-transit service and inet-transit service are close, and the difference of their median path inflation factors is within 0.2, except that the factor of Google's WAN-transit routes in the Middle East is 0.3 less than that of its inet-transit routes. To put these numbers in perspective, for every 1000 km distance between two locations, every 0.1 increment in the path inflation factor corresponds to about a 1 ms increment in RTT. In addition, we observe that Google's routing in Asia (Fig. 6a) has large path inflation factors, indicating a potential for improvement which we explore in §5.3. For the Middle East region, shown in Fig. 6c, both Azure's WAN-transit and inet-transit services do not perform as well as the other clouds. Furthermore, by comparing Fig. 6d with the other figures, we find that all three clouds achieve very efficient routing in North America. As for the other three regions not shown in the figure, all three clouds show similar efficiency, where the differences between the path inflation factors are all less than 0.2 (Appendix E).

In summary, from the perspective of median values of path inflation factors, among 12 combinations of clouds and regions shown above, six of them show that the WAN-transit service is more efficient than the inet-transit service, and in the other four combinations, WAN-transit and inet-transit achieve the same path efficiency. Finally, inet-transit service is more efficient than WAN-transit in two combinations (AWS in Europe and Google in Asia).

5.2 Latency Jitter

Besides latency values, the latency variations are also important for users' quality of experience for Internet services such as video chat and online gaming. Therefore, we also compute the RTT



(a) Asia (b) Europe (c) Middle East (d) North America

Fig. 7. CDFs of the *latency jitter* for ⟨probe, VM⟩ pairs. The numbers in the legend indicate the median values of the *jitter* of each service.

jitter for each cloud's WAN-transit and inet-transit service. For each ⟨probe, VM⟩ pair, we define the RTT jitter as the difference between the 95th percentile RTT and the median RTT [47, 55].

Fig. 7 shows the CDF of RTT jitter of <probe, VM> pairs of each cloud's network services. We show the results for the same four regions as in Fig. 6 and present the jitter of other regions in Appendix F. Jitter within 30–50 ms is considered acceptable for different applications [2, 27, 50]. As shown in the figure, both WAN-transit and inet-transit services have a jitter less than 20 ms for more than 90% of the ⟨probe, VM⟩ pairs for clouds and regions, well within the acceptable jitter range. When we examine the median jitter values, a cloud's WAN-transit service usually has the same or a slightly smaller median jitter value with three exceptions: AWS's and Azure's Asia regions and Azure's Middle East region. In addition, Google's inet-transit service has a 3.1 ms larger median jitter than its WAN-transit service in the Middle East region, suggesting that jitter reduction is a main benefit of its WAN-transit service in this region.

5.3 Alternative Routing Strategies

From previous sections, we observe that WAN-transit traffic does not always egress the cloud providers' network at the cloud edges closest to the destination. Moreover, in some cloud regions, existing services have low route efficiency, *e.g.* Google in Asia (Fig. 6a), Azure in Middle East, and AWS in Europe (Fig. 6b). These observations motivate us to explore how alternative routing strategies can reduce end-to-end latencies.

We emulate three alternative routing strategies using the methods described in §3:

`Lowest-Latency`: We emulate performance-aware routing by synthesizing a route between a ⟨probe, VM⟩ pair such that the sum of the two segment's RTTs, between the probe to its cloud edge and between the cloud edge to the VM, is the lowest among the WAN-transit, the inet-transit, and all alternative routes we synthesize. We pre-screen the candidate lowest latency paths as described earlier in §3.4.

`Closest-Edge`: We synthesize an alternative route between a ⟨probe, VM⟩ pair using the closest cloud edge to the probe for both the ingress and egress directions. If it has lower latency than the default cloud route, then we switch to this route for the ⟨probe, VM⟩ pair. This approach explores only one alternative route's latency and is more lightweight than the lowest-latency approach.

`Shortest-Distance`: We synthesize an alternative route between a ⟨probe, VM⟩ pair such that the sum of the geodesic distances from a probe–cloud edge and from the cloud edge–VM is the shortest among all alternative routes we can synthesize. We switch to this route if it has lower latency.

Recall that when we construct an alternative route, we require a probe previously ingressed the cloud via the same AS as the alternative cloud edge in a traceroute output, as this condition signals that the alternative route is a valid path. We count the available alternative cloud edges

(identified by ⟨ASN, metro⟩) for each probe in our dataset, and we find that about 75% of probes have alternative cloud edges. This is because, all cloud providers we study offer two types of services, thereby often offering two alternative ingress ASes for each probe.

For the `Closest-Edge` and `Shortest-Distance` categories, a chosen edge may include multiple policy-compliant IPs for a probe. We use the IP address that provides the lowest synthesized RTT as the ping target to construct the alternative route.

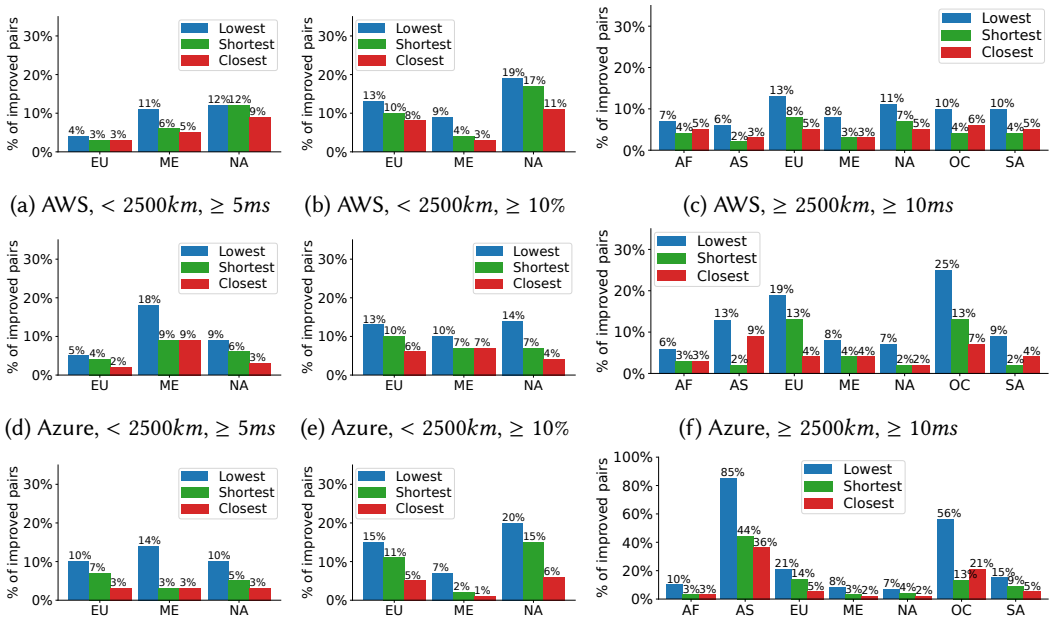
Metrics of Improvement: How much latency reduction is significant? Surprisingly, there does not seem to be a definite answer to this question. We derive two thresholds from the literature to quantify whether an alternative route significantly reduces the path latency between a ⟨probe, VM⟩ pair [1, 53, 66]. First, for web applications we consider latency reduction significant if an alternative route reduces the path latency by 10 ms. This threshold is based on an Akamai report where a 100 ms increase in website load time can decrease e-commerce sales by 7% [1], and a recent study shows that 10 is a reasonable lower bound on the number of round trips required for loading a web page [53]. Second, when a probe is within a short distance (<2500 km) of a cloud region, we consider a 5 ms and 10% RTT reduction as significant. This is because within such a short distance, a cloud region can support interactive applications that have a stringent latency requirement such as interactive online gaming. For such applications, a latency larger than 100 ms typically results in poor QoE [37]. Therefore, we consider a 5 ms or 10% RTT reduction as significant. Admittedly, these two thresholds cannot fully describe the improvement of alternative routing strategies. Thus, we also present the CDF of RTT reduction for different cloud regions in Appendix G (Figs. 14–20) for completeness.

When we compare the RTT difference between an alternative route and a default cloud route, we apply the median RTT difference with confidence metric [25, 73, 79]. Specifically, for each ⟨probe, VM⟩ pair, let η_c or η_a denote the median RTT value from our measurements for a cloud or an alternative route, respectively. We then compute a 95% confidence interval $[I_{left}\ ,\ I_{right}]$ such that the median RTT reduction $\eta_c - \eta_a$ is within this interval with 95% probability [73]. We consider an alternative route achieves a 5 ms (or 10 ms) latency reduction only if the left edge of the interval I_{left} is larger than 5 ms (or 10 ms). We compute the percentage of RTT reduction: $\frac{\eta_c - \eta_a}{\eta_c}$ in the same manner.

Fig. 8 shows the alternative route strategies' latency reduction results when compared to a cloud's WAN-transit service. Results compared to inet-transit services are similar and are omitted for brevity. We find that `Lowest-Latency` routing achieves the most improvement as it explores alternative routes extensively. In contrast, `Closest-Edges` routing usually provides the least improvement, while `Shortest-Distance` is in the middle. These two strategies are less effective because they do not directly take latency into consideration when making alternative routing choices, and neither `Closest-Edge` nor Shortest-Distance is a reliable indicator of lowest latency.

Next, we dive into the results from `Lowest-Latency` routing in more detail. As shown in Fig. 8 (a, d, g), for ⟨probe, VM⟩ pairs less than 2500 km apart, the percentage of pairs with more than 5 ms reduction by `Lowest-Latency` routing are 4%–10% for Europe, 11%–18% for Middle East and 9%–12% for North America for different clouds. We do not have results for other regions because there are too few probes (< 100) within the distance range (< 2500 km) to those regions.

Europe appears to have the lowest percentage of improved pairs by the absolute latency reduction value (5 ms). However, if we examine Fig. 8 (b, e, h)), which depicts the percentage of ⟨probe, VM⟩ pairs whose RTT reduction exceeds 10%, we find a larger percentage of improved pairs, ranging from 13% to 15% in different clouds. This is because the majority of RIPE Altas probes are located in Europe and their RTTs to the Europe VM are small compared to those to other regions.



(a) AWS, $< 2500km$, $\geq 5ms$ (b) AWS, $< 2500km$, $\geq 10\%$ (c) AWS, $\geq 2500km$, $\geq 10ms$

(d) Azure, $< 2500km$, $\geq 5ms$ (e) Azure, $< 2500km$, $\geq 10\%$ (f) Azure, $\geq 2500km$, $\geq 10ms$

(g) Google, $< 2500km$, $\geq 5ms$ (h) Google, $< 2500km$, $\geq 10\%$ (i) Google, $\geq 2500km$, $\geq 10ms$

Fig. 8. The bars show the percentages of ⟨probe, VM⟩ pairs which have the RTT reduction with the confidence interval's lower bound larger than the thresholds when comparing three alternative strategies to WAN-transit service. The numbers above the bars indicate the concrete percentage. The x-axis shows the abbreviations of regions according to Table 1. Some regions are missing in the figures of "$< 2500km$" because too few ⟨probe, VM⟩ pairs are included in that range.

Moreover, the `Lowest-Latency` routes can still shorten median RTTs by 10% for 14–20% of probes' traffic traveling to North America, although all three clouds' current WAN-transit services are already very efficient in North America (see §5.1).

As for ⟨probe, VM⟩ pairs more than 2500 km apart (Fig. 8 (c, f, i)), we also observe significant improvement of more than 10 ms provided by the `Lowest-Latency` routing in all three clouds. Firstly, AWS has such an improvement for 7%-13% ⟨probe, VM⟩ pairs in all seven regions. We also find more than 19% of ⟨probe, VM⟩ pairs benefit from 10 ms RTT reduction for traffic to Azure's Europe and Oceania regions. Notably, Google's Asia region has the most improved ⟨probe, VM⟩ pairs, where the `Lowest-Latency` alternative routes reduce the RTTs for 85% of ⟨probe, VM⟩ pairs by 10 ms—where more than 60% of the ⟨probe, VM⟩ pairs' RTTs can be improved by more than 100 ms, and the largest median RTT reduction is more than 200 ms (see Fig. 15 in Appendix G). This result is also consistent with our finding of low route efficiency for Google in the Asia region in §5.1. When taking a closer look at the traceroute data to Google's Asia region, we find that Google usually routes the traffic from probes in Europe to the VM in Asia (Mumbai, India) through the Atlantic and Pacific Oceans, while the `Lowest-Latency` routes use the path through the Eurasia continent, significantly reducing the travel distance by more than 10,000 km. Prior work also observed a similar detoured Google route [25].

Summary: Using both the median path inflation metric and the RTT jitter metric, for three cloud providers across four regions, we find that the performance of a cloud's WAN-transit service is comparable or slightly better to that of its inet-transit service with a few exceptions. In addition,

we find that the `Lowest-Latency` routing strategy can significantly reduce median RTTs in regions where cloud routing is inefficient.

6 SUMMARY OF TEMPORAL VALIDATION

We conducted two identical measurement campaigns five months apart (§3.5) to obtain two sets of measurements. We found that the patterns we observed in the first persist in the second set of measurements. The results presented in the main body of this work are primarily from the later measurements, but we present all relevant figures for the earlier measurements in Appendix H.

Specifically, we find that cloud providers continue to employ diverse routing strategies, including global anycast, regional anycast, and unicast to provide WAN-transit and inet-transit services. Except for Azure's Middle East region (§4.1), all other cloud regions' routing strategies remain the same. Consequently, the inet-transit and WAN-transit services' latency distributions remain stable, with the median path inflation factors changing within 0.3 for all cloud providers and all regions. Finally, the latency reductions from alternative routing strategies remain similar across the two measurement campaigns. For instance, for ⟨probe, VM⟩ pairs more than 2500 km apart, the alternative Lowest-Latency routing strategy continued to reduce the latency of Google's Asia region by 100 ms for more than 60% of the pairs.

7 LIMITATIONS AND FUTURE WORK

Possible geolocation errors: Although we have carefully designed the IP geolocation method (§3.2), possible errors may still exist such as inaccurate results from Geofeeds and the Geo-IP database. However, we believe these errors do not affect the overall conclusions of this paper. First, we use coarse-grained geolocation (>500 km) in our analysis (§ 4 and §5), and the geolocation method we used is highly accurate at this granularity as shown in §3.2. Second, when relatively fine-grained geolocation (<50 km) is used for edge detection (Fig. 1b), we reduce potential errors using an RTT difference filter (<2 ms) and speed of light violation (Fig. 1b). We also aggregate and limit the edge geolocations to the cloud providers' published PoP locations. We plan to systematically evaluate the accuracy of our IP geolocation method in our future work.

Other Metrics: This study aims to understand cloud routing strategies for tiered network services and their impacts on end-to-end latencies. We do not measure other performance metrics such as throughput, packet loss rate, or reliability. Large-scale and accurate measurements of such metrics require resources we currently do not have. For instance, RIPE Atlas does not support throughput measurements to user-defined destinations and measuring packet loss rates requires more measurement traffic over a longer period of time [47, 70]. We defer developing cost-effective measurement methods for these metrics to future work.

Median RTT: We only use the median RTT to compare alternative and existing routes, which cannot capture other properties such as the long tail and minimum delay. We focus on the median RTT because it reflects the normal-case user experience, and unlike the average, it is robust against outliers caused by failures, outages, path changes, and extreme congestion. Additionally, median RTT/latency is also adopted by prior studies [25, 35, 37, 47, 91, 92]. Referring to prior work [47], we also investigate RTT jitters for WAN-transit and inet-transit routes (§5.2). We leave the exploration of other percentiles for alternative paths' RTTs or their jitters to future work.

Other Cloud Providers: We limit our study to the three largest cloud providers, which account for 66% of the cloud infrastructure service market worldwide according to Statista [6]. Our conclusions may not extend to other smaller cloud providers that do not build their own global private WANs.

RIPE Atlas: RIPE Atlas has a skewed distribution [82] with most probes concentrated in North America and Europe (Appendix A). We expect our conclusions to hold to a large degree if we were to

use vantage points at other locations, because we have uncovered a diverse set of routing strategies (§4) and cloud providers likely use them across their infrastructure. We leave the confirmation with other research platforms such as distributed VPNs [87] in the future work.

IPv6 Traffic: Our current measurement focuses on IPv4 traffic as the majority of the Internet users still use IPv4 to access the Internet, according to Google [13] and Cloudflare [75]. Future work can adopt our measurement methodology to study cloud routing strategies and latencies for IPv6.

Longitudinal Study: We conducted two measurement campaigns five months apart and observed that the main findings hold over this time span. Future work can periodically repeat the measurement and observe how cloud routing strategies and performance change over time.

Performance of application traffic: In this paper, we only measure the path latency in the network layer. It is unclear whether application traffic would be routed in a path differing from the measurement traffic, and whether the alternative paths provide latency improvement for application traffic. We plan to investigate the routing strategies for application specific traffic in the future work.

Egress Change Detection for inet-transit Traffic: Finally, for at least one cloud provider, egress locations of inet-transit traffic may change when the traffic volume increases [16]. A interesting future direction is to study how to trigger the changes and how the changes impact performance.

8 RELATED WORK

Prior Cloud Performance Studies: Arnold et al. measured and compared the median latency of AWS' and Google's WAN-transit and inet-transit services [25]. This work extends the previous work in the following ways. First, Arnold et al.'s work did not explore the potential alternative paths, except for only a few outliers with exceptionally poor performance were analyzed. In contrast, we develop a systematic method to investigate how alternative routing strategies may improve WAN-transit traffic's latencies. Second, we more thoroughly study both the ingress and egress strategies—in particular the overlapping routing strategies for inet-transit and WAN-tranist—of the cloud providers, while Arnold et al.'s work primarily focused on where the Internet traffic enters the cloud, *i.e.*, the ingress direction. Third, our measurement is at a larger scale as the cloud providers have greatly expanded their operations, consisting of three cloud providers (AWS, Azure, and Google) and seven cloud regions for each provider, while the previous work studied two cloud providers (AWS and Google) and three regions for each provider.

Yeganeh et al. [90] studied the latency and throughput of inter-cloud connections between U.S. East coast and West coast using three connectivity methods: the clouds' private WANs, the public Internet, and a third-party connectivity provider. Their work shows that cloud WANs have the best latency and throughput performance for inter-cloud connections. This work studies the latencies between Internet hosts world wide and cloud VMs, complementing the earlier works' findings.

Mok et al. measured the throughput performance from VMs on Google Cloud to selected speed test servers [67], while we use the latency metric to study the routing strategies of three largest cloud providers. Haq et al. [47] compared the packet loss rate, latency jitter, and available bandwidth of 22 inter-continental intra-cloud paths with those of the Internet paths. They found that intra-cloud paths have lower packet loss rates, lower tail latency jitter (95 percentile and beyond), and higher available bandwidth. Unlike their work, ours has a much larger scale. We measure the latencies of more than 200k paths between Internet end systems and cloud VMs, and studies the impacts of routing strategies on such latencies.

Wang et al. [85] compared the performance of two types of cloud providers: one built by connecting datacenters with a private WAN and the other built by building datacenters using an existing ISP backbone. Their study is limited to three cloud providers that are popular in China:

Alibaba, Tencent, and CTYun. Similar to this work, their work focuses on latency comparisons. They, however, measure datacenter and traffic egress locations by sending measurement traffic from VMs to the Internet and do not employ Internet vantage points. Consequently, when they study alternative cloud routing strategies, they employ a what-if analysis with approximated latencies. In contrast, this work experimentally measures alternative route latencies and sends measurement traffic from both an Internet vantage point and a cloud VM to a shared intermediary router.

Pi et al. [71] showed that load-balancers inside a cloud's WAN or in the Internet could affect the measured minimum path latency, and they use the 10^{th} percentile to eliminate anomalous traffic. Miao et al. [65] also had a similar observation. We chose to use the median latency metric to offset the effects of load balancers, as it filters out the outlier values resulted from potentially overloaded load balancers [55]. Additionally, we do not use the cloud provider's load balancing options, so our measurement techniques should only be affected by the cloud providers' and ISPs' traffic engineering efforts.

Alternative paths: A previous measurement study [81] showed that either early or late exit policy between two ISPs leads to optimal latency performance. This work in part confirmed this finding in the context of cloud routing, as we show that we can synthesize lower median latency routes by entering or exiting a cloud's network via alternative cloud edges. Detour [78] and RON [68] showed that routing via an intermediary Internet host can provide better performance that the default routes between two hosts. J-QoS [46] and XRON [86] constructed cloud overlay paths to optimize performance while minimizing WAN-transit costs, while OverQoS [83] used Internet hosts as overlay nodes to improve applications' QoS. PAINTER [54] exposes alternative paths to reach a cloud region by strategically advertising different IP prefixes at different PoPs and to different peers. A client can select the best path to reach a cloud region by selecting IP addresses. Differently, this work constructs alternative paths to reach a cloud VM using policy-compliant cloud edges uncovered through traceroute experiments.

AS Border Detection and Geolocation: Our work builds upon the vast body of work that detects AS borders and geolocates the border IP addresses [26, 42, 43, 56, 59, 62, 63, 77, 89] to identify the locations where traffic enters or leaves a cloud. We used multiple publicly available sources and developed a ranked algorithm to geo-locate where traffic enters or leaves a cloud's WAN.

9 CONCLUSION

In this work, we conduct a large scale measurement on the routing strategies, the resulting access latencies, and the potential approaches for reducing access latencies of three large cloud providers' tiered network services. Our study shows that the routing strategies these service providers adopt do not precisely match the advertised service descriptions. For instance, some of their WAN-transit traffic exits the cloud WANs early, traveling thousands of kilometers of extra distance outside the WANs. We further show that both the WAN-transit and the inet-transit services can experience less than satisfactory access latencies, and a performance-based routing scheme can significantly reduce the access latencies. In our experiments, we observe more than 100 ms latency reductions in some cloud regions for a subset of our vantage points.

ACKNOWLEDGMENTS

We sincerely thank our shepherd Gareth Tyson and the anonymous reviewers for their constructive comments. We gratefully thank the RIPE Atlas community for their credit donation and measurement support. This work is supported in part by NSF awards CNS-1901523, CNS-2148275, and CNS-2225448 and by Google Cloud Research Credits Program.

REFERENCES

[1] 2017. Akamai Online Retail Performance Report: Milliseconds are Critical. Retrieved in Apr, 2024 from https://www.ir.akamai.com/news-releases/news-release-details/akamai-online-retail-performance-report-milliseconds-are.

[2] 2020. What is Jitter? - Cisco Meraki Documentation. Retrieved in Oct, 2024 from https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/What_is_Jitter%3F.

[3] 2022. IP2Geolocation. Retrieved in Dec, 2022 from https://www.ip2location.com/.

[4] 2022. MaxMind. Retrieved in Dec, 2022 from https://www.maxmind.com/en/home.

[5] 2023. Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $600 Billion in 2023. Retrieved in April, 2024 https://www.gartner.com/en/newsroom/press-releases/2023-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023.

[6] 2024. Amazon Maintains Cloud Lead as Microsoft Edges Closer. https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/.

[7] 2024. AWS Global Accelerator Features. Retrieved in April, 2024 from https://aws.amazon.com/global-accelerator/features/.

[8] 2024. AWS IP Address Ranges. Retrieved in May, 2024 from https://docs.aws.amazon.com/vpc/latest/userguide/aws-ip-ranges.html.

[9] 2024. Azure Front Door POP Locations by Metro. Retrieved in Jan, 2024 from https://learn.microsoft.com/en-us/azure/frontdoor/edge-locations-by-region.

[10] 2024. Azure IP Ranges and Service Tags - Public Cloud. Retrieved in May, 2024 from https://www.microsoft.com/en-us/download/details.aspx?id=56519.

[11] 2024. Hurricane Electric Internet Exchange Report. Retrieved in May, 2024 from https://bgp.he.net/report/exchanges.

[12] 2024. IPinfo. Retrieved in Apr, 2024 from https://ipinfo.io/.

[13] 2024. IPv6-Google. Retrieved in Oct, 2024 from https://www.google.com/intl/en/ipv6/statistics.html.

[14] 2024. Network edge locations. Retrieved in Jan, 2024 from https://cloud.google.com/vpc/docs/edge-locations.

[15] 2024. Network Service Tiers. Retrieved in April, 2024 from https://cloud.google.com/network-tiers.

[16] 2024. Network Service Tiers Overview. Retrieved in May, 2024 from https://cloud.google.com/network-tiers/docs/overview#traffic_routing.

[17] 2024. Obtain Google IP Address Ranges. Retrieved in May, 2024 from https://support.google.com/a/answer/10026322?hl=en.

[18] 2024. OpenGeoFeed. Retrieved in Apr, 2024 from https://opengeofeed.org/.

[19] 2024. Packet Clearing House Internet Exchange Point Datasets. Retrieved in May, 2024 from https://www.pch.net/ixp/data.

[20] 2024. Peering and Interconnection. Retrieved in Jan, 2024 from https://aws.amazon.com/peering/.

[21] 2024. PeeringDB. Retrieved in May, 2024 from https://www.peeringdb.com/.

[22] 2024. RouteViews. Retrieved in Sep, 2024 from http://www.routeviews.org/routeviews/.

[23] 2024. What is Routing Preference? Retrieved in April, 2024 from https://learn.microsoft.com/en-us/azure/virtual-network/ip-services/routing-preference-overview.

[24] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. 2015. Investigating Interdomain Routing Policies in the Wild. In Proceedings of ACM IMC'15. ACM, 71–77.

[25] Todd Arnold, Ege Gürmeriçliler, Georgia Essig, Arpit Gupta, Matt Calder, Vasileios Giotsas, and Ethan Katz-Bassett. 2020. (How Much) Does a Private WAN Improve Cloud Performance?. In Proceedings of IEEE INFOCOM'20. IEEE, 79–88.

[26] Todd Arnold, Jia He, Weifan Jiang, Matt Calder, Italo Cunha, Vasileios Giotsas, and Ethan Katz-Bassett. 2020. Cloud Provider Connectivity in the Flat Internet. In Proceedings of ACM IMC'20. ACM, 230–246.

[27] Anastasiia Beznosyk, Peter Quax, Karin Coninx, and Wim Lamotte. 2011. Influence of Network Delay and Jitter on Cooperation in Multiplayer Games. In Proceedings of International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI'11). ACM, 351–354.

[28] Marjory Blumenthal, Ramesh Govindan, Ethan Katz-Bassett, Arvind Krishnamurthy, James McCauley, Nick Merrill, Tejas Narechania, Aurojit Panda, and Scott Shenker. 2024. Can We Save the Public Internet? ACM SIGCOMM Computer Communication Review (CCR) 53, 3 (2024), 18–22.

[29] R Bonica, D Gan, D Tappan, and C Pignataro. 2007. ICMP Extensions for Multiprotocol Label Switching. RFC 4950. Internet Engineering Task Force (IETF). 1–8 pages. https://www.rfc-editor.org/info/rfc4950

[30] Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2017. Why is the Internet So Slow?!. In Proceedings of Springer PAM'17. Springer, 173–187.

[31] Randy Bush, Massimo Candela, Warren Kumari, and Russ Housley. 2021. Finding and Using Geofeed Data. RFC 9092. Internet Engineering Task Force (IETF). 1–21 pages. https://www.rfc-editor.org/info/rfc9092
[32] CAIDA. 2024. CAIDA Internet eXchange Points (IXPs) Dataset. Retrieved in May, 2024 from https://www.caida.org/catalog/datasets/ixps/.
[33] Massimo Candela. 2023. geofeed-finder. Retrieved in Nov, 2023 from https://github.com/massimocandela/geofeed-finder.
[34] Kelvin Chan. 2023. Meta fined record $1.3 billion and ordered to stop sending European user data to US. https://apnews.com/article/meta-facebook-data-privacy-fine-europe-9aa912200226c3d53aa293dca8968f84.
[35] Lorenzo Corneo, Maximilian Eder, Nitinder Mohan, Aleksandr Zavodovski, Suzan Bayhan, Walter Wong, Per Gunningberg, Jussi Kangasharju, and Jörg Ott. 2021. Surrounded by the Clouds: A Comprehensive Cloud Reachability Study. In Proceedings of ACM WWW'21. ACM, 295–304.
[36] Leslie Daigle. 2004. Whois Protocol Specification. RFC 3912. Internet Engineering Task Force (IETF). 1–4 pages. https://www.rfc-editor.org/info/rfc3912
[37] The Khang Dang, Nitinder Mohan, Lorenzo Corneo, Aleksandr Zavodovski, Jörg Ott, and Jussi Kangasharju. 2021. Cloudy with a Chance of Short RTTs: Analyzing Cloud Connectivity in the Internet. In Proceedings of ACM IMC'21. ACM, 62–79.
[38] Omar Darwich, Hugo Rimlinger, Milo Dreyfus, Matthieu Gouel, and Kevin Vermeulen. 2023. Replication: Towards a Publicly Available Internet Scale IP Geolocation Dataset. In Proceedings of ACM IMC'23. ACM, 1–15.
[39] Ben Du, Massimo Candela, Bradley Huffaker, Alex C Snoeren, and KC Claffy. 2020. RIPE IPmap Active Geolocation: Mechanism and Performance Evaluation. ACM SIGCOMM Computer Communication Review (CCR) 50, 2 (2020), 3–10.
[40] Lixin Gao and Jennifer Rexford. 2001. Stable Internet routing without global coordination. IEEE/ACM Transactions on Networking 9, 6 (2001), 681–692.
[41] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A Look at Router Geolocation in Public and Commercial Databases. In Proceedings of ACM IMC'17. ACM, 463–469.
[42] Vasileios Giotsas, Thomas Koch, Elverton Fazzion, Ítalo Cunha, Matt Calder, Harsha V. Madhyastha, and Ethan Katz-Bassett. 2020. Reduce, Reuse, Recycle: Repurposing Existing Measurements to Identify Stale Traceroutes. In Proceedings of ACM IMC'20. ACM, 247–265.
[43] Vasileios Giotsas, Georgios Smaragdakis, Bradley Huffaker, Matthew Luckie, and kc claffy. 2015. Mapping Peering Interconnections at the Facility Level. In Proceedings of ACM CoNEXT 2015. ACM, 1–13.
[44] Hadi Asghari. 2023. pyasn. Retrieved in Feb, 2023 from https://pypi.org/project/pyasn/.
[45] Shuai Hao, Yubao Zhang, Haining Wang, and Angelos Stavrou. 2018. End-Users Get Maneuvered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks. In Proceedings of USENIX Security'18. USENIX, 1129–1145.
[46] Osama Haq, Cody Doucette, John W. Byers, and Fahad R. Dogar. 2020. Judicious QoS using Cloud Overlays. In Proceedings of ACM CoNEXT'20. ACM, 371–385.
[47] Osama Haq, Mamoon Raja, and Fahad R. Dogar. 2017. Measuring and Improving the Reliability of Wide-Area Cloud Paths. In Proceedings of ACM WWW'17. ACM, 253–262.
[48] Yihua He, Michalis Faloutsos, Srikanth Krishnamurthy, and Bradley Huffaker. 2005. On Routing Asymmetry in the Internet. In Proceedings of IEEE Globecom'05, Vol. 2. IEEE, 904–909.
[49] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. 2013. Achieving high utilization with software-driven WAN. In Proceedings of ACM SIGCOMM'13. ACM, 15–26.
[50] Eben Howard, Clint Cooper, Mike P Wittie, Steven Swinford, and Qing Yang. 2014. Cascading Impact of Lag on Quality of Experience in Cooperative Multiplayer Games. In Proceedings of Annual Workshop on Network and Systems Support for Games (NetGames'14). IEEE, 1–6.
[51] Ethan Katz-Bassett, Harsha V Madhyastha, Vijay Kumar Adhikari, Colin Scott, Justine Sherry, Peter Van Wesep, Thomas E Anderson, and Arvind Krishnamurthy. 2010. Reverse Traceroute. In Proceedings of USENIX NSDI'10. USENIX, 219–234.
[52] Erik Kline, Krzysztof Duleba, Zoltan Szamonek, Stefan Moser, and Warren Kumari. 2020. A Format for Self-Published IP Geolocation Feeds. RFC 8805. Internet Engineering Task Force (IETF). 1–23 pages. https://www.rfc-editor.org/info/rfc8805
[53] Thomas Koch, Ke Li, Calvin Ardi, Ethan Katz-Bassett, Matt Calder, and John Heidemann. 2021. Anycast in Context: A Tale of Two Systems. In Proceedings of ACM SIGCOMM'21. ACM, 398–417.
[54] Thomas Koch, Shuyue Yu, Sharad Agarwal, Ethan Katz-Bassett, and Ryan Beckett. 2023. PAINTER: Ingress Traffic Engineering and Routing for Enterprise Cloud Networks. In Proceedings of ACM SIGCOMM'23. ACM, 360–377.
[55] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. 2013. Detecting Outliers: Do not Use Standard Deviation Around the Mean, Use Absolute Deviation Around the Median. Journal of Experimental Social Psychology 49, 4 (2013), 764–766.

[56] Ioana Livadariu, Kevin Vermeulen, Maxime Mouchet, and Vasilis Giotsas. 2024. Geofeeds: Revolutionizing IP Geolocation or Illusionary Promises?. In Proceedings of ACM CoNEXT'24. ACM, 1–21.

[57] Natasha Lomash. 2024. Uber fined $324M over EU drivers' data transfer breach. Retrieved in Oct, 2024 from https://techcrunch.com/2024/08/26/uber-fined-324m-over-eu-driver-data-transfer-breach/.

[58] Matthew Luckie. 2010. Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet. In Proceedings of ACM IMC'10. ACM, 239–245.

[59] Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, David Clark, and KC Claffy. 2016. bdrmap: Inference of Borders Between IP Networks. In Proceedings of ACM IMC'16. ACM, 381–396.

[60] Matthew Luckie, Bradley Huffaker, Alexander Marder, Zachary Bischof, Marianne Fletcher, and K Claffy. 2021. Learning to Extract Geographic Information from Internet Router Hostnames. In Proceedings ACM CoNEXT'21. ACM, 440–453.

[61] Ritesh Maheshwari. 2015. TCP over IP Anycast - Pipe Dream or Reality? Retrieved in Sep, 2022 from https://engineering.linkedin.com/network-performance/tcp-over-ip-anycast-pipe-dream-or-reality.

[62] Alexander Marder, oKimberly C K.C. Claffy, and Alex C. Snoeren. 2021. Inferring Cloud Interconnections: Validation, Geolocation, and Routing Behavior. In Proceedings of Springer PAM'21. Springer, 230–246.

[63] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, KC Claffy, and Jonathan M Smith. 2018. Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale. In Proceedings of ACM IMC'18. ACM, 56–69.

[64] Stephen McQuistin, Sree Priyanka Uppu, and Marcel Flores. 2019. Taming Anycast in the Wild Internet. In Proceedings of ACM IMC'19. ACM, 165–178.

[65] Congcong Miao, Zhizhen Zhong, Yunming Xiao, Feng Yang, Senkuo Zhang, Yinan Jiang, Zizhuo Bai, Chaodong Lu, Jingyi Geng, Zekun He, et al. 2024. MegaTE: Extending WAN Traffic Engineering to Millions of Endpoints in Virtualized Cloud. In Proceedings of ACM SIGCOMM'24. ACM, 103–116.

[66] Nitinder Mohan, Lorenzo Corneo, Aleksandr Zavodovski, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. 2020. Pruning Edge Research with Latency Shears. In Proceedings of ACM HotNets. ACM, 182–189.

[67] Ricky K. P. Mok, Hongyu Zou, Rui Yang, Tom Koch, Ethan Katz-Bassett, and K C Claffy. 2021. Measuring the Network Performance of Google Cloud Platform. In Proceedings of ACM IMC'21. ACM, 54–61.

[68] Akihiro Nakao, Larry Peterson, and Andy Bavier. 2006. Scalable Routing Overlay Networks. ACM SIGOPS Operating Systems Review 40, 1 (2006), 49–61.

[69] RIPE NCC. 2023. RIPE Database. Retrieved in Nov, 2023 from https://www.ripe.net/manage-ips-and-asns/db/.

[70] Vern Paxson. 1997. End-to-end Internet Packet Dynamics. In Proceedings of ACM SIGCOMM'97. ACM, 139–152.

[71] Yibo Pi, Sugih Jamin, Peter Danzig, and Feng Qian. 2020. Latency Imbalance Among Internet Load-Balanced Paths: A Cloud-Centric View. Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS'20) 4, 2 (2020), 1–29.

[72] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. 2011. IP Geolocation Databases: Unreliable? ACM SIGCOMM Computer Communication Review (CCR) 41, 2 (2011), 53–56.

[73] Robert M Price and Douglas G Bonett. 2002. Distribution-free Confidence Intervals for Difference and Ratio of Medians. Journal of Statistical Computation and Simulation 72, 2 (2002), 119–124.

[74] Yakov Rekhter and Tony Li. 1994. A Border Gateway Protocol 4 (BGP-4). RFC 4271. Internet Engineering Task Force (IETF). 1–104 pages. https://www.rfc-editor.org/info/rfc4271

[75] Carlos Rodrigues. 2023. Using DNS to Estimate the Worldwide State of IPv6 Adoption. Retrieved in Oct, 2024 https://blog.cloudflare.com/ipv6-from-dns-pov/.

[76] Eric Rosen, Arun Viswanathan, and Ross Callon. 2001. Multiprotocol Label Switching Architecture. RFC 3031. Internet Engineering Task Force (IETF). 1–61 pages. https://www.rfc-editor.org/info/rfc3031

[77] Loqman Salamatian, Todd Arnold, Ítalo Cunha, Jiangchen Zhu, Yunfan Zhang, Ethan Katz-Bassett, and Matt Calder. 2023. Who Squats IPv4 Addresses? ACM SIGCOMM Computer Communication Review (CCR) 53, 1 (2023), 48–72.

[78] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. 1999. Detour: informed Internet routing and transport. IEEE Micro 19, 1 (1999), 50–59.

[79] Brandon Schlinker, Italo Cunha, Yi-Ching Chiu, Srikanth Sundaresan, and Ethan Katz-Bassett. 2019. Internet Performance from Facebook's Edge. In Proceedings of ACM IMC'19. ACM, 179–194.

[80] Job Snijders. 2016. PeeringDB Accuracy: Is Blind Faith Reasonable? NANOG58.

[81] Neil Spring, Ratul Mahajan, and Thomas Anderson. 2003. The Causes of Path Inflation. In Proceedings of ACM SIGCOMM'03. ACM, 113–124.

[82] RIPE NCC Staff. 2015. RIPE Atlas: A Global Internet Measurement Network. Internet Protocol Journal 18 (2015).

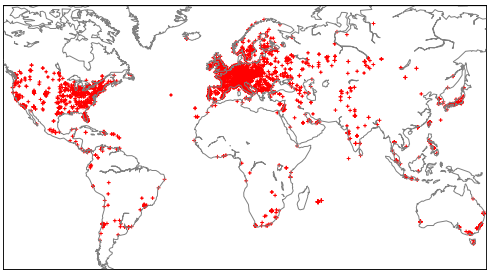
[83] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy Katz. 2004. OverQoS: An Overlay Based Architecture for Enhancing Internet QoS. In Proceedings of USENIX NSDI'04. USENIX, 71–84.

[84] IPinfo Team. 2024. How accurate is IPinfo's IP address location: verifying IP data accuracy. Retrieved in April 2024 https://ipinfo.io/blog/verifying-ip-address-accuracy/.

[85] Qinkai Wang, Ye Tian, Xin Yu, Lan Ding, and Xinming Zhang. 2023. Where is the Traffic Going? A Comparative Study of Clouds Following Different Designs. IEEE Transactions on Services Computing 16, 2 (2023), 1473–1484.
[86] Bingyang Wu, Kun Qian, Bo Li, Yunfei Ma, Qi Zhang, Zhigang Jiang, Jiayu Zhao, Dennis Cai, Ennan Zhai, Xuanzhe Liu, et al. 2023. XRON: A Hybrid Elastic Cloud Overlay Network for Video Conferencing at Planetary Scale. In Proceedings of ACM SIGCOMM'23. ACM, 696–709.
[87] Yunming Xiao, Matteo Varvello, and Aleksandar Kuzmanovic. 2022. Monetizing Spare Bandwidth: The Case of Distributed VPNs. Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS'22) 6, 2 (2022), 1–27.
[88] 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 Proceedings of ACM SIGCOMM'17. ACM, 432–445.
[89] Bahador Yeganeh, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. 2019. How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric. In Proceedings of ACM IMC'19. ACM, 202–216.
[90] Bahador Yeganeh, Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. 2020. A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet. In Proceedings of Springer PAM'20. Springer, 193–210.
[91] Xiao Zhang, Tanmoy Sen, Zheyuan Zhang, Tim April, Balakrishnan Chandrasekaran, David Choffnes, Bruce M. Maggs, Haiying Shen, Ramesh K. Sitaraman, and Xiaowei Yang. 2021. AnyOpt: Predicting and Optimizing IP Anycast Performance. In Proceedings of ACM SIGCOMM'21. ACM, 447–462.
[92] Minyuan Zhou, Xiao Zhang, Shuai Hao, Xiaowei Yang, Jiaqi Zheng, Guihai Chen, and Wanchun Dou. 2023. Regional IP Anycast: Deployments, Performance, and Potentials. In Proceedings of ACM SIGCOMM'23. ACM, 917–931.

A GEOGRAPHICAL DISTRIBUTION OF PROBES

Fig. 9 shows the footprint of the probes used in our measurements. We have 65.9% of probes located in Europe and 19.0% of probes located in North America. In contrast, the probes in other continents are all less than 10%.



(a) Geographical distribution

Continent	#Probes	Percentage
Europe	3432	65.9%
North America	987	19.0%
Asia	427	8.2%
Oceania	157	3.0%
Africa	109	2.1%
South America	93	1.8%
Total	5205	100.0%

(b) Number of chosen probes by continent

Fig. 9. Geographical distribution of RIPE Atlas probes used in our measurements: (a) the locations of probes (b) the number and percentage of probes grouped by continent.

B VALIDATION OF GROUPING PROBES BY <ASN, CITY>

In this work, we use one random probe from each <ASN, city> group to save the cost, because we expect that the the probes in the same <ASN, city> group will enter the cloud through the same cloud edges. We validate this argument in this section.

We randomly select 200 groups of <ASN, city> in our dataset, and we randomly select 5 probes from each group. Then, we use these $200 \times 5 = 1000$ probes to launch traceroutes through both WAN-transit and inet-transit services to all seven regions of three clouds used in our study, including $2 \times 7 \times 3 - 1 = 41$ destination IP addresses (Google's South Africa region does not provide inet-transit service). We group the traceroute by a tuple of <destination IP, ASN, city>, and thus, each group of <destination IP, ASN, city> contains five probes and corresponding traceroutes. We apply our

edge discover method (§ 3.2) on the traceroutes. Since we may fail on cloud edge discovery for a traceroute because of unresponsive hops, we finally have 4216 groups of <destination IP, ASN, city> where each group contains at least two probes and corresponding detected cloud edges from their traceroutes. If we find that the cloud edges detected in the probes' traceroutes of a <destination IP, ASN, city> group are all in the same city (or within 50 km), we consider this <destination IP, ASN, city> group has consistent cloud edges. Then we find that 3892 (92%) of all 4216 groups have consistent cloud edges. Since we resolve the cloud edge at the metro level, we conclude that the probes from the same <ASN, city> group will take the same cloud edge to enter a cloud's WAN.

C LOOKING GLASS VALIDATION

For Azure's Asia region, we examined 10 looking glass servers provided by 10 different ingress ASes that show up in our traceroutes most frequently to validate the presence and geolocation of our uncovered edge IP at different locations other than the dominant location. We ran BGP route query and traceroute from the looking glass server located at the exact same ingress location of our traceroute measurement.

Table 7. The ingress ASes whose looking glass servers we use to validate the presence and geolocations of the cloud-end edge IPs.

Ingress ASN	Cloud-end edge IP	Location
55850	43.243.22.38	Auckland, NZ
7713	104.44.42.19	Singapore, SG
9790	104.44.6.113	Auckland, NZ
134697	104.44.236.45	Sydney, AU
28792	104.44.53.151	Manchester, UK
7545	104.44.44.209	Sydney, AU
8262	193.169.198.74	Sofia, BG
224	104.44.198.71	Oslo, NO
37199	196.60.70.47	Johannesburg, ZA
48943	193.203.0.165	Vienna, AT

We inspect traceroute results from the RIPE Atlas probe and looking glass server to the VM's inet-transit address. Since the ingress AS is the intermediate hop before the traffic enters the cloud, we should be able to observe that both traceroutes are using the same cloud edge to reach the VM. On the other hand, the locations of the looking glass servers are published on the provider's website. If the edge hop has a RTT smaller than 2 ms, we can infer that it is co-located with the corresponding looking glass server, hence the geolocation is accurate. The list of looking glass servers and the edge IP we examined are shown in Table 7.

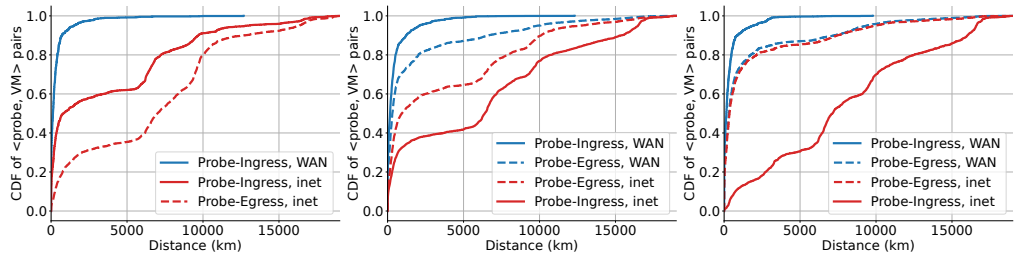
Within the 10 we examined, 7 of them we can directly observe the ingress cloud edge we uncovered from RIPE Atlas probe traceroute to the VM's inet-transit address through both BGP route query and traceroute from looking glass server. We can also validate that we have the correct geolocation for cloud edge since the location of looking glass server is given and the RTT from it to the cloud edge is within 2 ms.

For three of them (AS55850, AS9790, AS7545), the route from looking glass server to the VM's inet-transit address enters the cloud in a location different than the route from probe to inet-transit VM traceroute. Instead, we find that the if we traceroute from looking glass server to VM's WAN-transit address, we will observe the same cloud edge as the probe to inet-transit traceroute, and its RTT is also within 2 ms range.

D CDF OF DISTANCES

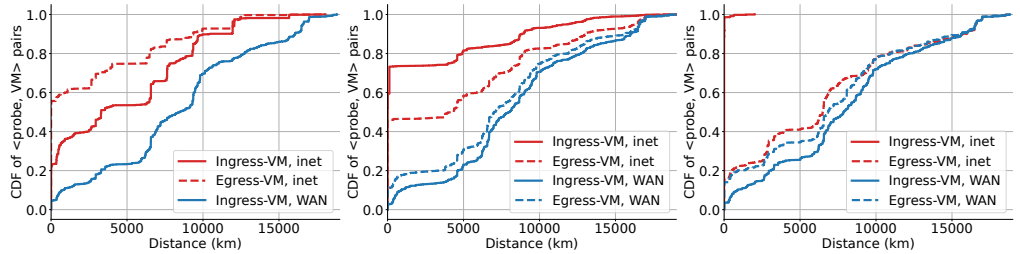
As shown in Fig. 10, we find that for 84% (AWS), 78% (Azure), and 81% (Google) of the WAN-transit ⟨probe, VM⟩ pairs, a probe enters a cloud provider's WAN within a distance of 500 km. This result shows that the global anycast strategy employed by cloud providers is highly effective in enabling early entry to a cloud's WAN. However, only 61%(Azure) and 61%(Google) of the egress traffic travels less than 500 km from a cloud edge to a probe.

Fig. 11 shows distance between the VM and the ingress (or egress) edges. We can find that for Azure, WAN-transit traffic (blue dashed curve) exits the edges further away than inet-transit traffic (red dashed curve). However, for Google, the blue and red dashed curves are close, indicating similar egress routing strategies of WAN-transit and inet-transit traffic, which is consistent with our observation in § 4.2 and § 4.3.



(a) AWS (b) Azure (c) Google

Fig. 10. Cumulative distributions of distance between a RIPE Atlas Probe and its ingress (or egress) cloud edge for both inet-transit and WAN-transit services across ⟨probe, VM⟩ pairs.



(a) AWS (b) Azure (c) Google

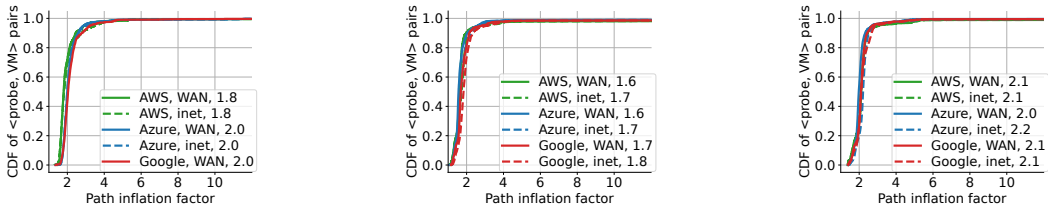
Fig. 11. Cumulative distribution of the distance between a ingress (or egress) cloud edge and the destination VM for both inet-transit and WAN-transit services across ⟨probe, VM⟩ pairs.

E PATH INFLATION FACTORS OF THREE OTHER REGIONS

Fig. 12 shows the path inflation factors of two network services for three cloud providers in the regions of Africa, Oceania, and South America. Overall, all three clouds show similar efficiency of WAN-transit and inet-transit services in these three regions.

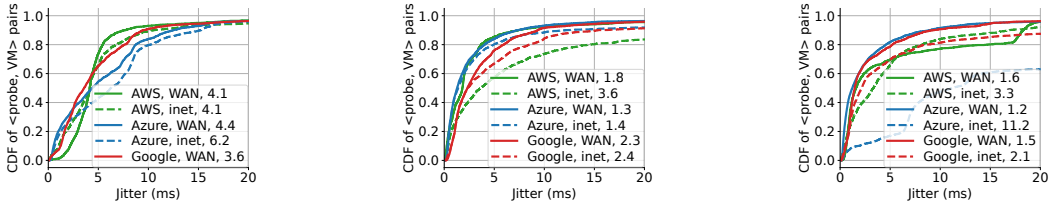
F LATENCY JITTERS OF THREE OTHER REGIONS

Fig. 13 shows the latency jitters of two network services for three cloud providers in the regions of Africa, Oceania, and South America.



(a) Africa (b) Oceania (c) South America

Fig. 12. CDFs of the path inflation factor for ⟨probe, VM⟩ pairs. The numbers in the legend indicate the median values of the path inflation factor of each service.

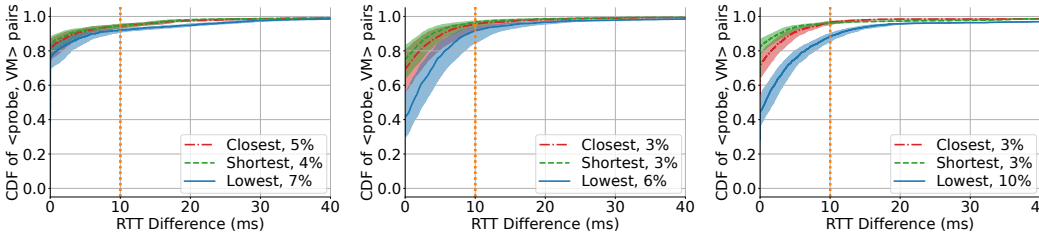


(a) Africa (b) Oceania (c) South America

Fig. 13. CDFs of the latency jitters for ⟨probe, VM⟩ pairs. The numbers in the legend indicate the median values of the jitter of each service. Google does not provide WAN-transit service in its Africa region.

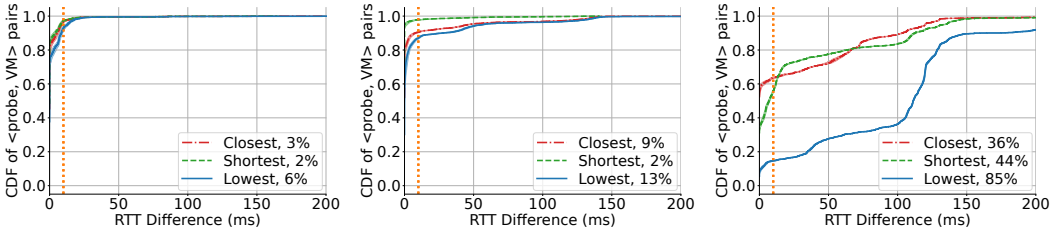
G CDF OF THE RTT DIFFERENCE (PERCENTAGE) WITH CONFIDENCE INTERVAL

The following figures (Fig. 14–20) show the CDF of the RTT difference (percentage) for difference cloud regions in the second measurement. Each figure includes one region of three clouds separated by column, and each row in the figure represents different groups, *i.e.* ⟨ RTT difference, $< 2500km$ ⟩, ⟨ RTT difference percentage, $< 2500km$ ⟩, and ⟨ RTT difference, $\geq 2500km$ ⟩. In the figures of Africa, Asia, Oceania, and South America, the results of the distance between ⟨probe, VM⟩ less than 2500 km are omitted due to the small number of ⟨probe, VM⟩ pairs.



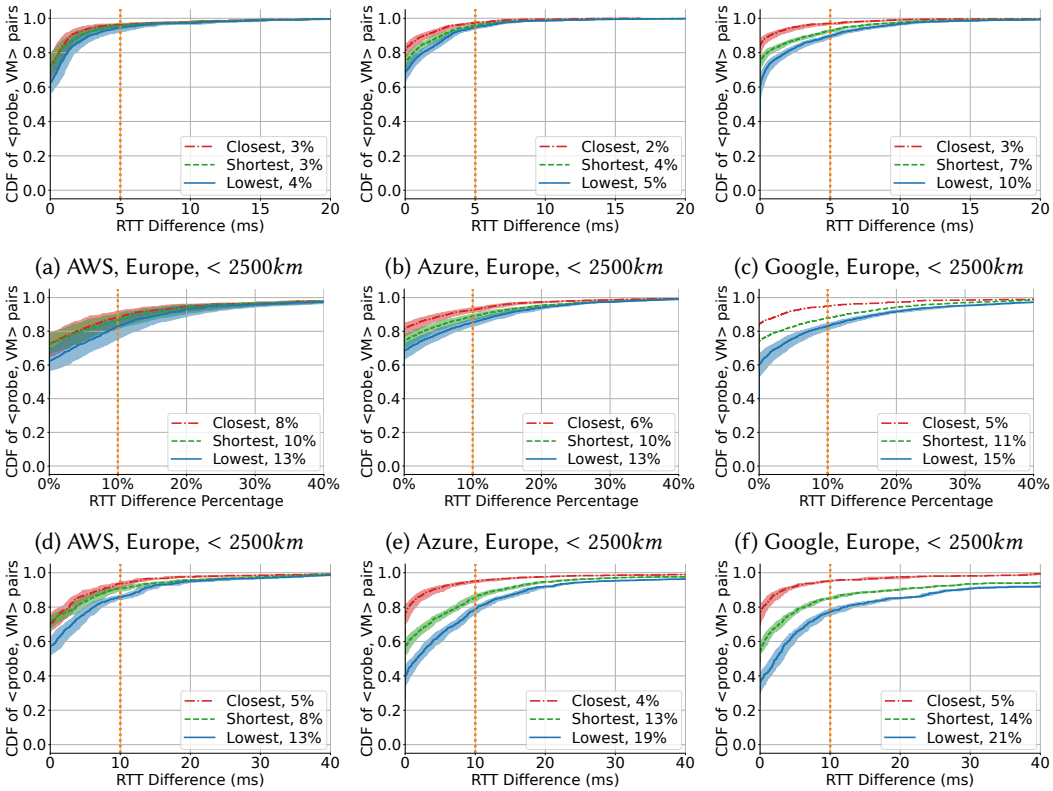
(a) AWS, Africa, $\geq 2500km$ (b) Azure, Africa, $\geq 2500km$ (c) Google, Africa, $\geq 2500km$

Fig. 14. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The shaded strips show 95% confidence interval $[I_{left}, I_{right}]$ of the median RTT difference. The orange line indicates the threshold of significant improvement (10 ms), and the numbers in the legends show the corresponding percentile in the CDF of I_{left}. *This figure shows the results of the second measurement (Table 5).*



(a) AWS, Asia, $\geq 2500km$ (b) Azure, Asia, $\geq 2500km$ (c) Google, Asia, $\geq 2500km$

Fig. 15. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The orange line indicates the threshold of significant improvement (10 ms). The legend definition is the same as Fig. 14. *This figure shows the results of the second measurement (Table 5).*

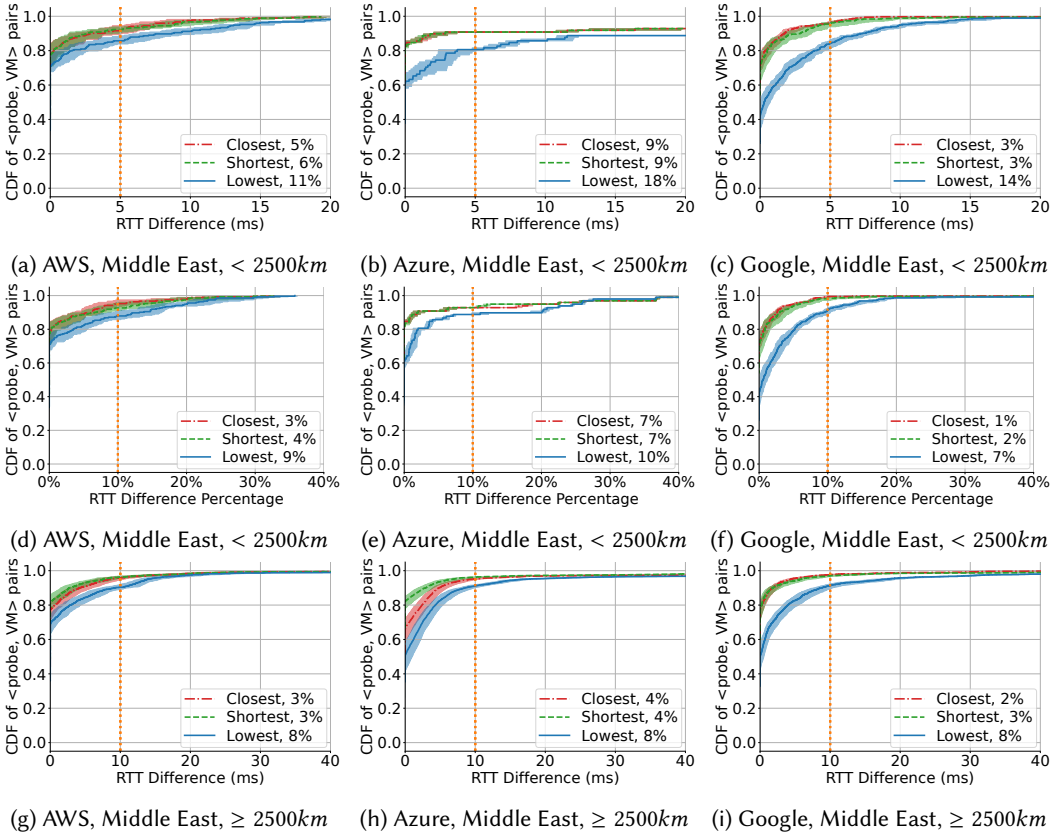


(a) AWS, Europe, $< 2500km$ (b) Azure, Europe, $< 2500km$ (c) Google, Europe, $< 2500km$

(d) AWS, Europe, $< 2500km$ (e) Azure, Europe, $< 2500km$ (f) Google, Europe, $< 2500km$

(g) AWS, Europe, $\geq 2500km$ (h) Azure, Europe, $\geq 2500km$ (i) Google, Europe, $\geq 2500km$

Fig. 16. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ less than 2500 km (a)-(f) or larger than 2500 km (g)-(i). The orange line indicates the threshold of significant improvement (5 ms, 10 ms, or 10%). The legend definition is the same as Fig. 14. *This figure shows the results of the second measurement (Table 5).*

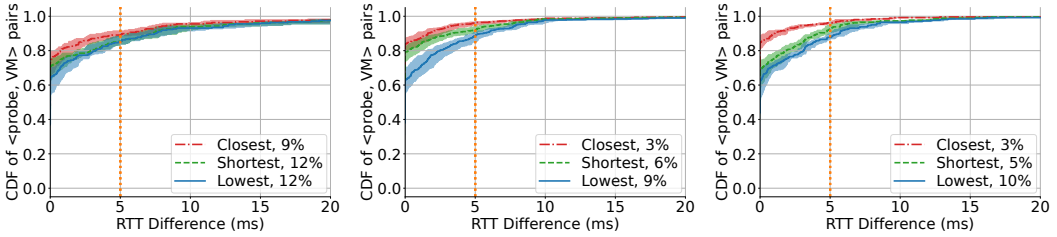


(a) AWS, Middle East, $< 2500km$ (b) Azure, Middle East, $< 2500km$ (c) Google, Middle East, $< 2500km$

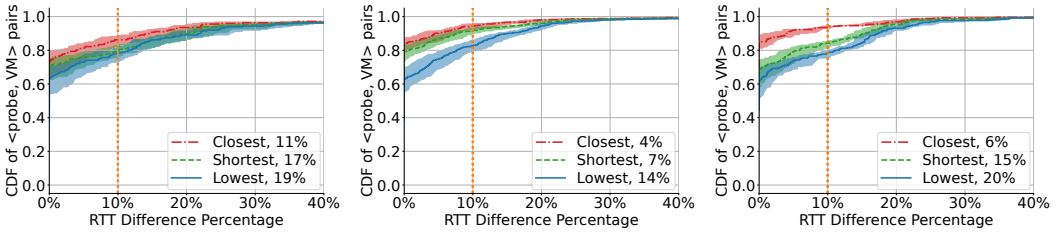
(d) AWS, Middle East, $< 2500km$ (e) Azure, Middle East, $< 2500km$ (f) Google, Middle East, $< 2500km$

(g) AWS, Middle East, $\geq 2500km$ (h) Azure, Middle East, $\geq 2500km$ (i) Google, Middle East, $\geq 2500km$

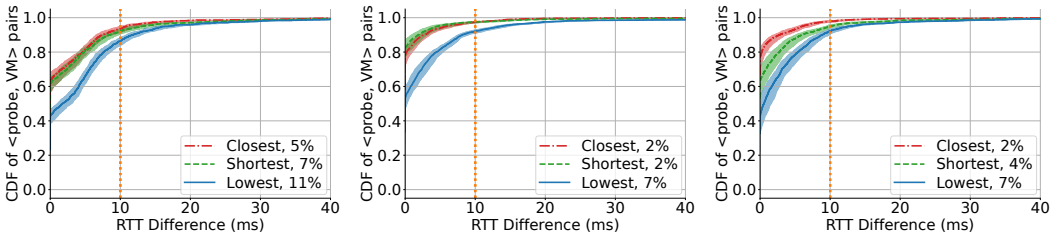
Fig. 17. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ less than 2500 km (a)-(f) or larger than 2500 km (g)-(i). The orange line indicates the threshold of significant improvement (5 ms, 10 ms, or 10%). The legend definition is the same as Fig. 14. *This figure shows the results of the second measurement (Table 5).*



(a) AWS, North America, $< 2500km$ (b) Azure, North America, $< 2500km$ (c) Google, North America, $< 2500km$

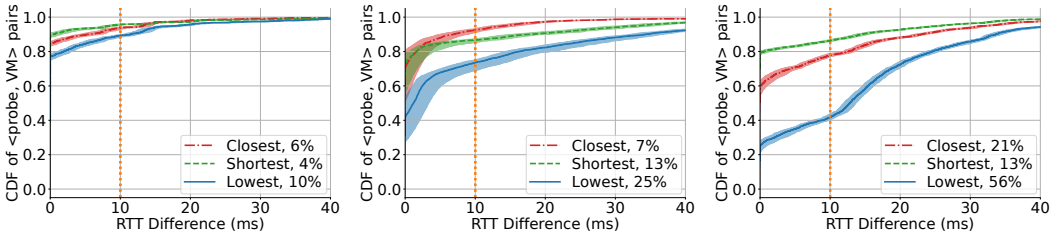


(d) AWS, North America, $< 2500km$ (e) Azure, North America, $< 2500km$ (f) Google, North America, $< 2500km$



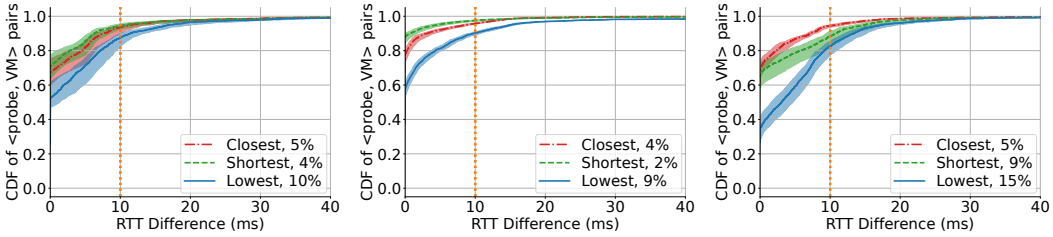
(g) AWS, North America, $\geq 2500km$ (h) Azure, North America, $\geq 2500km$ (i) Google, North America, $\geq 2500km$

Fig. 18. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ less than 2500 km (a)-(f) or larger than 2500 km (g)-(i). The orange line indicates the threshold of significant improvement (5 ms, 10 ms, or 10%). The legend definition is the same as Fig. 14. *This figure shows the results of the second measurement (Table 5).*



(a) AWS, Oceania, $\geq 2500km$ (b) Azure, Oceania, $\geq 2500km$ (c) Google, Oceania, $\geq 2500km$

Fig. 19. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The orange line indicates the threshold of significant improvement (10 ms). The legend definition is the same as Fig. 14. *This figure shows the results of the second measurement (Table 5).*



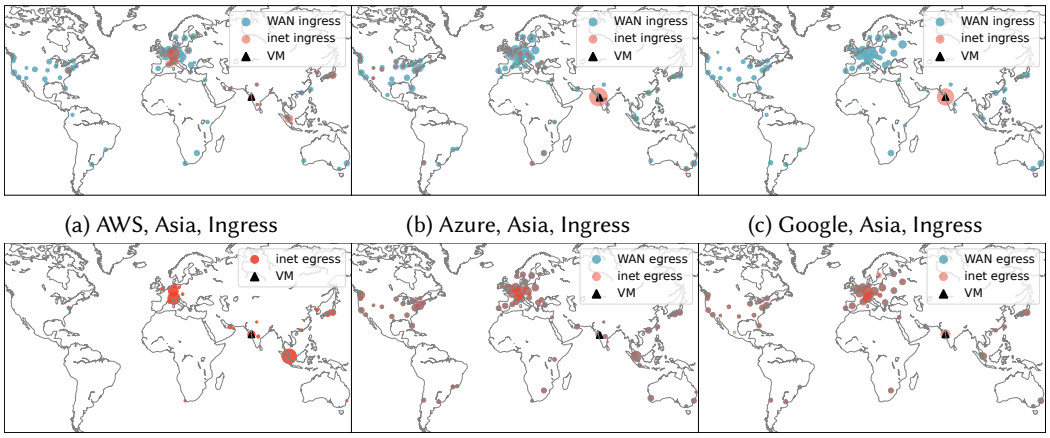
(a) AWS, South America, > 2500km (b) Azure, South America, > 2500km(c) Google, South America, > 2500km

Fig. 20. CDFs of median RTT reduction for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The orange line indicates the threshold of significant improvement (10 ms). The legend definition is the same as Fig. 14. *This figure shows the results of the second measurement (Table 5).*

H RESULTS OF THE FIRST MEASUREMENT

To ensure that the conclusions presented in our paper are persistent instead of transient, we conduct all measurements and analysis in the paper twice, which are separated by around five months as described in §3.5. In previous sections, we have presented the latest results in the second measurement, and in this section, we review and summarize the results and conclusions aforementioned, and we compare them with those in the first measurement. For completeness, we show all figures of corresponding results in the first measurement in the following section.

H.1 Ingress and Egress Cloud Edges

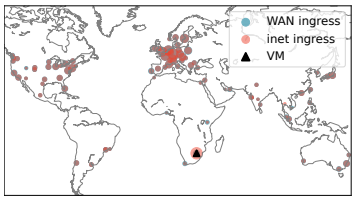


(a) AWS, Asia, Ingress (b) Azure, Asia, Ingress (c) Google, Asia, Ingress

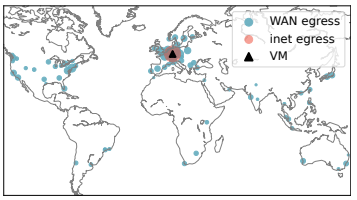
(d) AWS, Asia, Egress (e) Azure, Asia, Egress (f) Google, Asia, Egress

Fig. 21. Geographical distributions of the ingress/egress cloud edges for RIPE Atlas probes to reach different network service tiers of the Asia region for three cloud providers. The size of a circle indicates the number of probes ingressing/egressing the cloud edge. The egress edges of AWS WAN-transit service is not shown because AWS does not allow traceroutes from the cloud to the Internet through its WAN-transit service. *This figure shows the results of the first measurement (Table 5).*

Our findings of the routing strategies of each cloud in §4.1 and §4.2 are consistent between the two measurements. The geographical distribution of the old measurement is included in Fig. 21 and Fig. 22. The ingress and egress edges of AWS and Google for VMs in each region show the

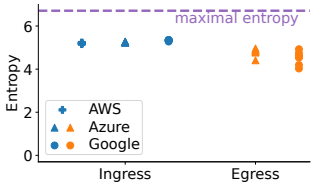


(a) Azure, Africa, Ingress

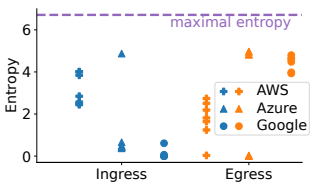


(b) Azure, Europe, Egress

Fig. 22. Geographical distributions of the ingress/egress cloud edges for RIPE Atlas probes to reach different network service tiers of the Africa and Europe regions for Azure. The legend definition is the same as Fig. 21. *This figure shows the results of the first measurement (Table 5).*



(a) Entropy (WAN-transit)



(b) Entropy (inet-transit)

	WAN	inet
AWS	N/A	33%
Azure	54%	40%
Google	67%	17%

(c) Symmetry

Fig. 23. (a) and (b): The entropy metric that captures the ingress/egress edge diversity of ⟨probe, VM⟩ pairs. (c) Percentage of ⟨probe, VM⟩ pairs that use the same ingress and egress cloud edges. *This figure shows the results of the first measurement (Table 5).*

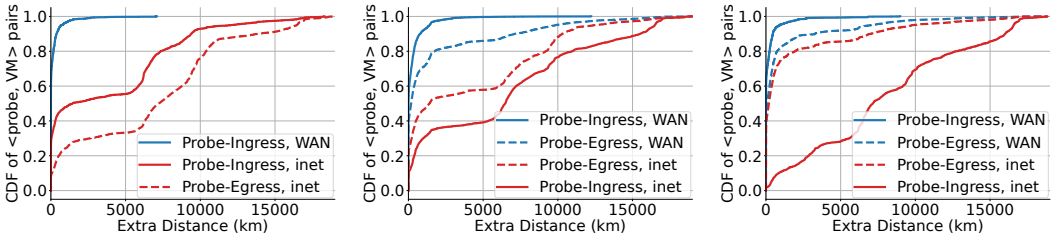
same distribution between two measurements. Azure's routing strategies also remain almost unchanged after five months except for the inet-transit in the Middle East, where the ingress edges are predominantly located in Europe in the first measurement and they are spread out to all around the world in the second measurement, as described in (§4.1).

H.2 Edge Diversity and Symmetry

We compare the entropy values of the ingress/egress routing strategies (§4.3). We find that the differences between the corresponding entropy values in the two measurements are within 0.42 for all regions except for Azure's inet-transit in the Middle East. As mentioned above, since the inet-transit prefix of the Azure's Middle East VM is announced globally in the second measurement, its entropy value rises from 0.40 to 5.3. The corresponding figures of the first measurement are included in Fig 23.

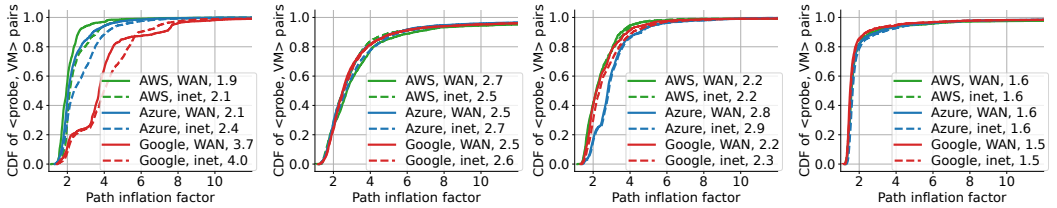
H.3 Extra Distances

For the extra distance to closest cloud edges caused by the cloud's routing strategies (§4.4), we find the percentages of ⟨probe, VM⟩ pairs entering or exiting the closest edge differ less than 4% for all clouds and network service tiers (Fig. 24) except for Azure's inet-transit ingress traffic. Due to the inet-transit routing changes of Azure in the Middle East, there are 7% more ⟨probe, VM⟩ pairs entering closet Azure edges. Combining all comparison results of routing strategies shown above, our conclusion of inconsistency between the cloud's actual routing strategies and their claims is persistent and reproducible.



(a) AWS (b) Azure (c) Google

Fig. 24. The cumulative distributions of the extra distance a probe travels compared to entering or leaving a cloud via its closest cloud edge for both WAN-transit and inet-transit services. We cannot obtain the egress edges for AWS's WAN-transit traffic. *This figure shows the results of the first measurement (Table 5).*



(a) Asia (b) Europe (c) Middle East (d) North America

Fig. 25. CDFs of the path inflation factor for ⟨probe, VM⟩ pairs. The numbers in the legend indicate the median values of the path inflation factor of each service. *This figure shows the results of the first measurement (Table 5).*

H.4 Path Inflation Factor

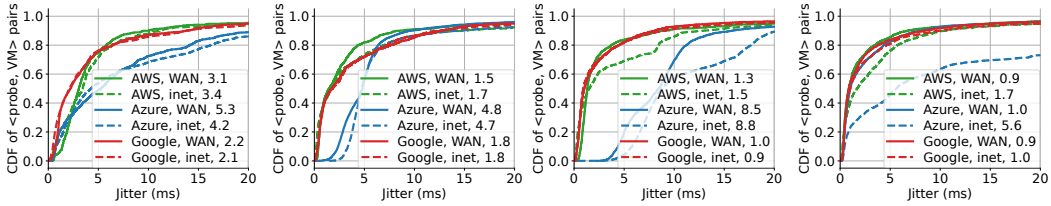
We compare the latency effect of the tiered routing. For the path inflation factor defined in §5.1, the differences of median values between the two measurements are less than 0.4 in all clouds and regions. Our conclusions such as the general efficiency of WAN-transit service, AWS's inefficient WAN transit to Europe, Google's inefficient WAN transit to Asia, and all clouds' efficient routings to North America are also valid in both time periods. The figures of the first measurement are included in Fig. 25.

H.5 Latency Jitters

Our observations on the latency jitter (§5.2) are also consistent in Fig. 26. A cloud's WAN-transit service does outperform its inet-transit service, but is usually less than 5 ms. Azure' Asia region shows a persistent exception that the WAN has a larger jitter than the Internet.

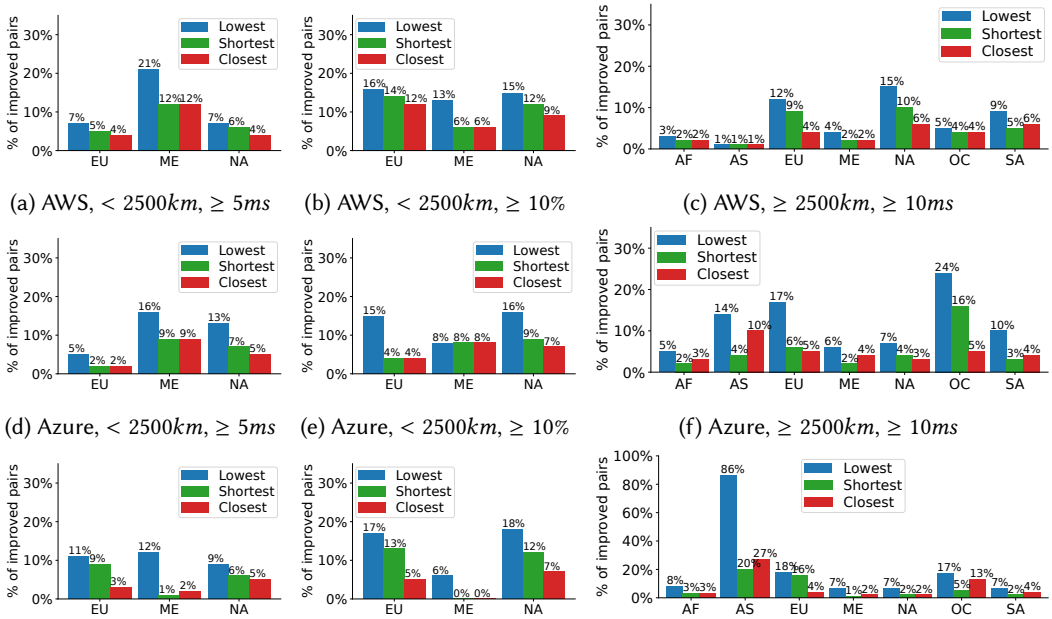
H.6 Alternative Routing Strategies

For the alternative routing strategies (§5.3), we find that `Closet-Edge` and `Shortest-Distance` routing still show less benefit of latency improvement than `Lowest-Latency` routing (Fig. 27). Besides, with measurements in different time periods, we can still find alternative routes that can improve the end-to-end latencies substantially. For example, in both measurements, the `Lowest-Latency` routes to Europe or North America can reduce more than 10% latency in more than 10% ⟨probe, VM⟩ pairs within 2500 km apart for all clouds. For ⟨probe, VM⟩ pairs more than 2500 km apart, both measurements show Azure can improve more than 15% of ⟨probe, VM⟩ pairs'



(a) Asia (b) Europe (c) Middle East (d) North America

Fig. 26. CDFs of the latency jitters for ⟨probe, VM⟩ pairs. The numbers in the legend indicate the median values of the path inflation factor of each service. *This figure shows the results of the first measurement (Table 5).*



(a) AWS, $< 2500km$, $\geq 5ms$ (b) AWS, $< 2500km$, $\geq 10\%$ (c) AWS, $\geq 2500km$, $\geq 10ms$

(d) Azure, $< 2500km$, $\geq 5ms$ (e) Azure, $< 2500km$, $\geq 10\%$ (f) Azure, $\geq 2500km$, $\geq 10ms$

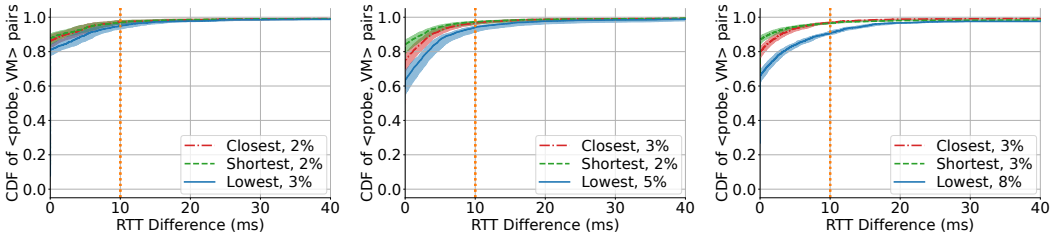
(g) Google, $< 2500km$, $\geq 5ms$ (h) Google, $< 2500km$, $\geq 10\%$ (i) Google, $\geq 2500km$, $\geq 10ms$

Fig. 27. The bars show the percentages of ⟨probe, VM⟩ pairs which have the RTT reduction with the confidence interval's lower bound larger than the thresholds when comparing three alternative strategies to WAN-transit service. The numbers above the bars indicate the concrete percentage. The x-axis shows the abbreviations of regions according to Table 1. Some regions are missing in the figures of "$< 2500km$" because too few ⟨probe, VM⟩ pairs are included in that range. *This figure shows the results of the second measurement (Table 5).*

latency by more than 10 ms for VMs located in Europe or Oceania. Besides, the `Lowest-Latency` routes persistently and significantly improve Google's routing in Asia, where more than 80% and 60% of ⟨probe, VM⟩ pairs can have more than 10 ms and 100 ms improvement, respectively. The improvement comes from the persistent path inflation issue in Google's routing for Asia as presented in §5.3. We also show the improvement CDFs and confidence intervals of the alternative routing strategies in Appendix H.7.

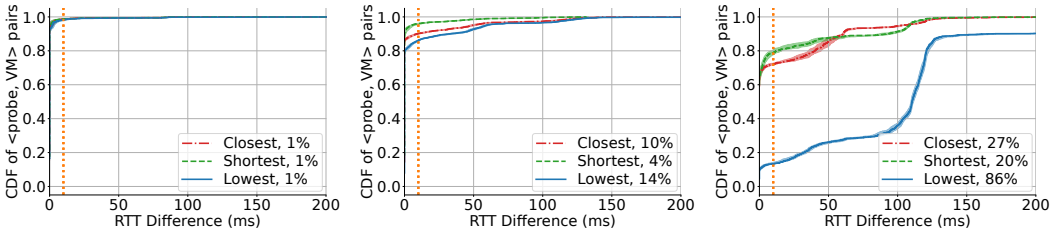
H.7 CDF of the RTT Difference (Percentage) with the Confidence Interval

The following figures (Fig. 28–34) show the CDF of the RTT difference (percentage) for difference cloud regions in the first measurement. Each figure includes one region of three clouds separated by column, and each row in the figure represents different groups, *i.e.* ⟨ RTT difference, $< 2500km$ ⟩, ⟨ RTT difference percentage, $< 2500km$ ⟩, and ⟨ RTT difference, $\geq 2500km$ ⟩. In the figures of Africa, Asia, Oceania, and South America, the results of the distance between ⟨probe, VM⟩ less than 2500 km are omitted due to the small number of ⟨probe, VM⟩ pairs.



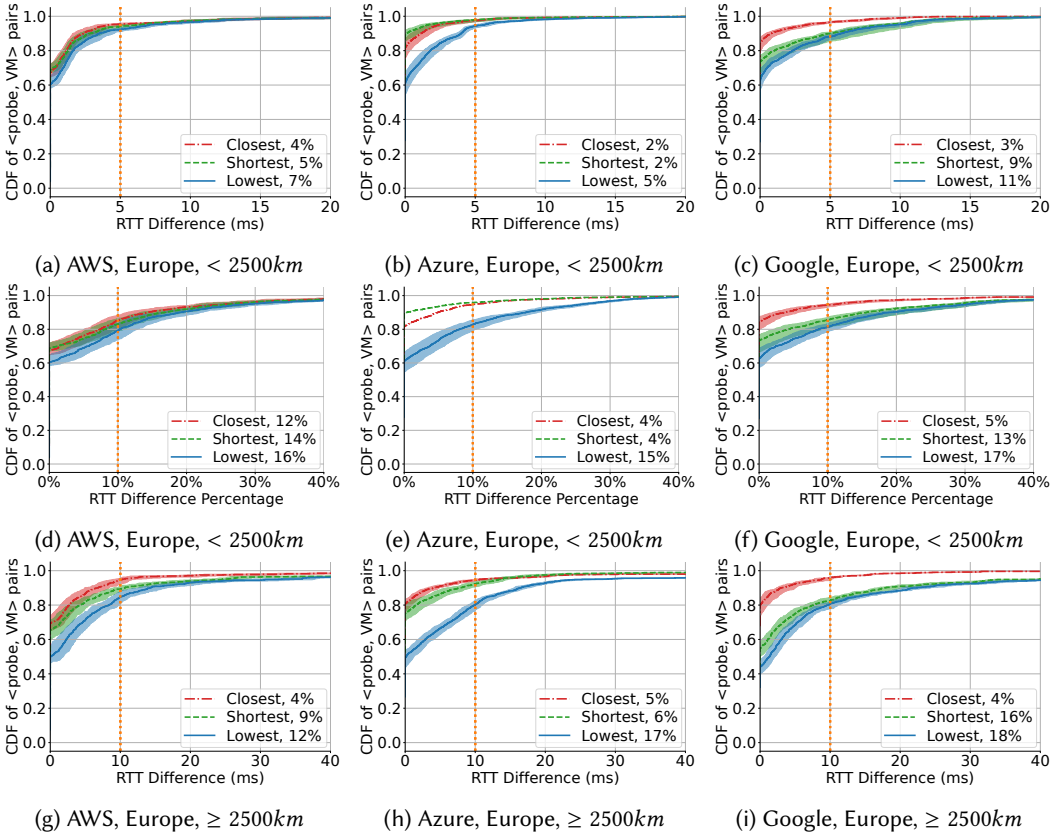
(a) AWS, Africa, $\geq 2500km$ (b) Azure, Africa, $\geq 2500km$ (c) Google, Africa, $\geq 2500km$

Fig. 28. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The shaded strips show 95% confidence interval $[I_{left}, I_{right}]$ of the median RTT difference. The orange line indicates the threshold of significant improvement (10 ms), and the numbers in the legends show the corresponding percentile in the CDF of I_{left}. *This figure shows the results of the first measurement (Table 5).*



(a) AWS, Asia, $\geq 2500km$ (b) Azure, Asia, $\geq 2500km$ (c) Google, Asia, $\geq 2500km$

Fig. 29. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The orange line indicates the threshold of significant improvement (10 ms). The legend definition is the same as Fig. 28. *This figure shows the results of the first measurement (Table 5).*

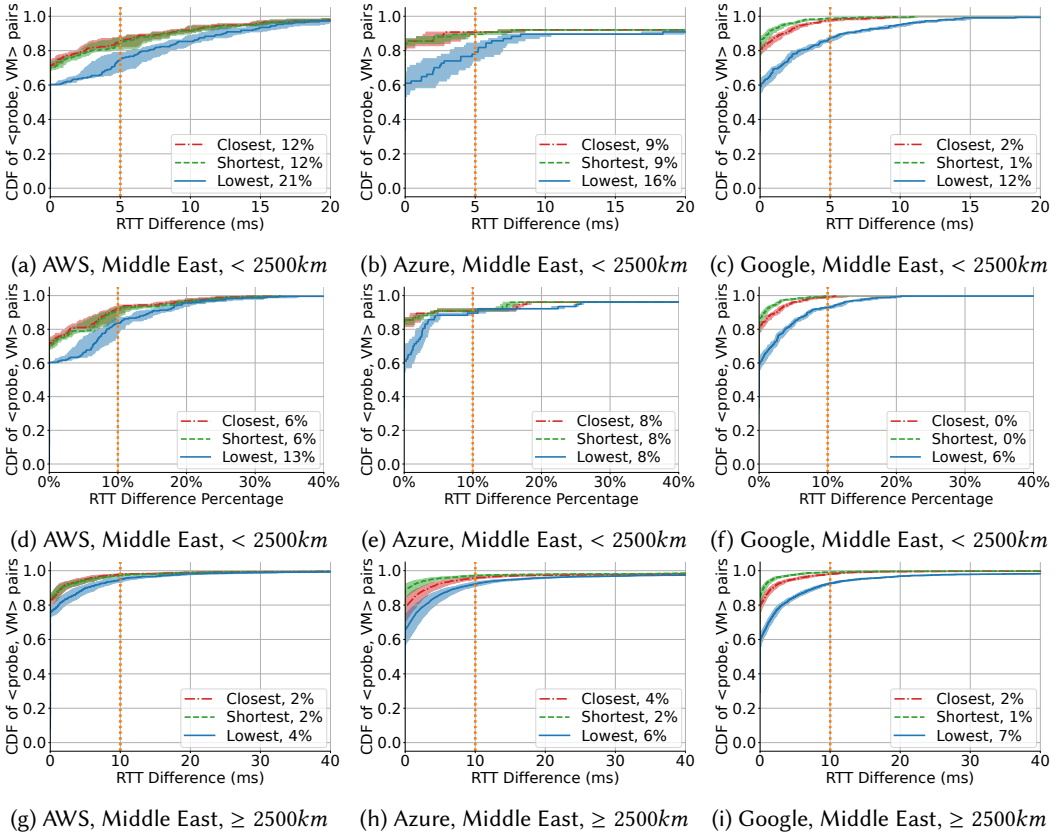


(a) AWS, Europe, $< 2500km$ (b) Azure, Europe, $< 2500km$ (c) Google, Europe, $< 2500km$

(d) AWS, Europe, $< 2500km$ (e) Azure, Europe, $< 2500km$ (f) Google, Europe, $< 2500km$

(g) AWS, Europe, $\geq 2500km$ (h) Azure, Europe, $\geq 2500km$ (i) Google, Europe, $\geq 2500km$

Fig. 30. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ less than 2500 km (a)-(f) or larger than 2500 km (g)-(i). The orange line indicates the threshold of significant improvement (5 ms, 10 ms, or 10%). The legend definition is the same as Fig. 28. *This figure shows the results of the first measurement (Table 5).*

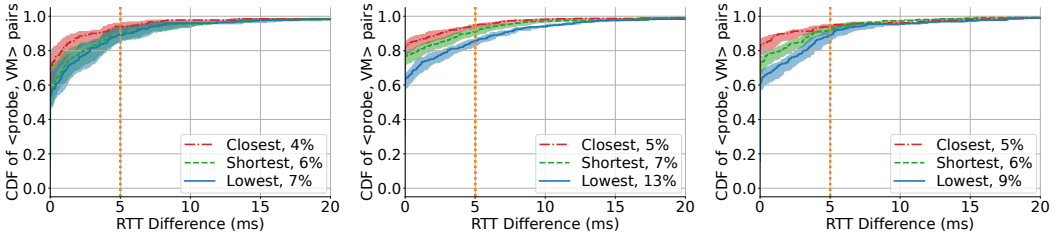


(a) AWS, Middle East, $< 2500km$ (b) Azure, Middle East, $< 2500km$ (c) Google, Middle East, $< 2500km$

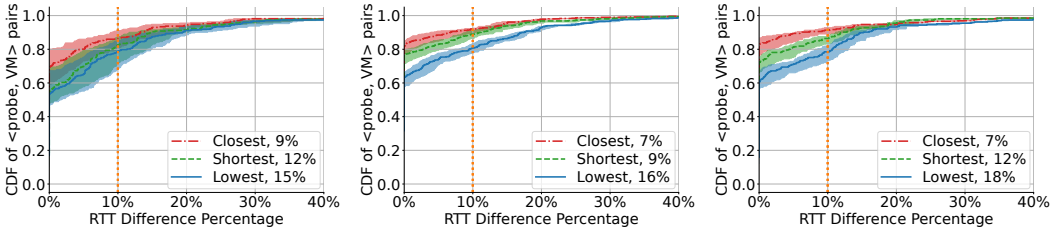
(d) AWS, Middle East, $< 2500km$ (e) Azure, Middle East, $< 2500km$ (f) Google, Middle East, $< 2500km$

(g) AWS, Middle East, $\geq 2500km$ (h) Azure, Middle East, $\geq 2500km$ (i) Google, Middle East, $\geq 2500km$

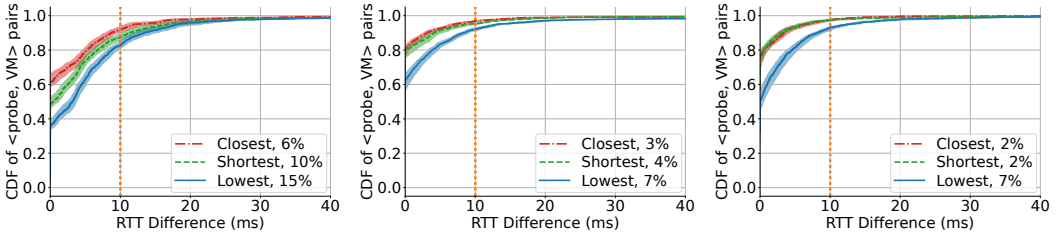
Fig. 31. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ less than 2500 km (a)-(f) or larger than 2500 km (g)-(i). The orange line indicates the threshold of significant improvement (5 ms, 10 ms, or 10%). The legend definition is the same as Fig. 28. *This figure shows the results of the first measurement (Table 5).*



(a) AWS, North America, $< 2500km$ (b) Azure, North America, $< 2500km$ (c) Google, North America, $< 2500km$

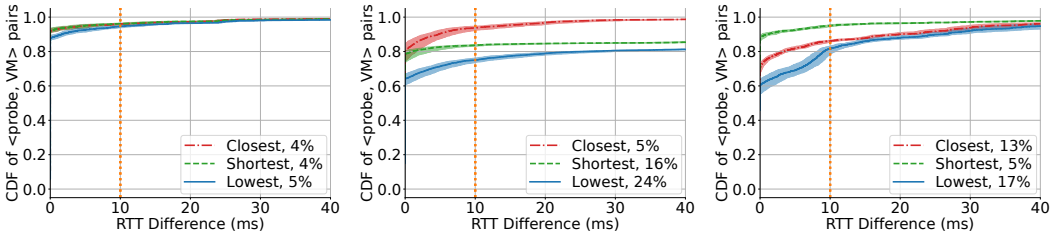


(d) AWS, North America, $< 2500km$ (e) Azure, North America, $< 2500km$ (f) Google, North America, $< 2500km$



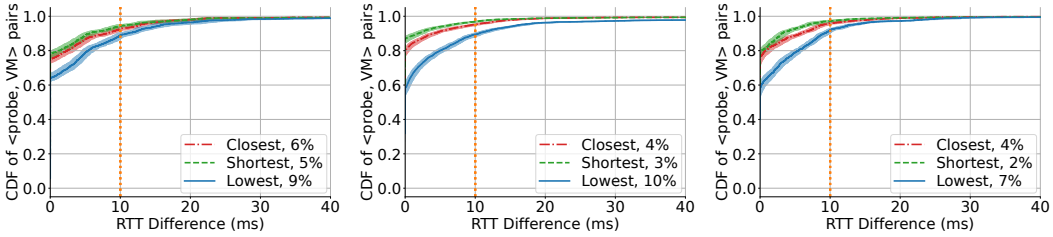
(g) AWS, North America, $\geq 2500km$ (h) Azure, North America, $\geq 2500km$ (i) Google, North America, $\geq 2500km$

Fig. 32. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ less than 2500 km (a)-(f) or larger than 2500 km (g)-(i). The orange line indicates the threshold of significant improvement (5 ms, 10 ms, or 10%) The legend definition is the same as Fig. 28. *This figure shows the results of the first measurement (Table 5).*



(a) AWS, Oceania, $\geq 2500km$ (b) Azure, Oceania, $\geq 2500km$ (c) Google, Oceania, $\geq 2500km$

Fig. 33. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The orange line indicates the threshold of significant improvement (10 ms). The legend definition is the same as Fig. 28. *This figure shows the results of the first measurement (Table 5).*



(a) AWS, South America, $\geq 2500km$ (b) Azure, South America, $\geq 2500km$(c) Google, South America, $\geq 2500km$

Fig. 34. CDFs of median RTT differences for alternative routes with the distance between ⟨probe, VM⟩ larger than 2500 km (a)-(c). The orange line indicates the threshold of significant improvement (10 ms). The legend definition is the same as Fig. 28. *This figure shows the results of the first measurement (Table 5).*

Received October 2024; revised January 2025; accepted January 2025