Tiered Cloud Routing: Methodology, Latency, and Improvement

Shihan Lin
Duke University
Durham, NC, USA

Yi Zhou
Duke University
Durham, NC, USA

Xiao Zhang*
Cisco ThousandEyes
Raleigh, NC, USA

Todd Arnold†
U.S. Military Academy
West Point, NY, USA

Ramesh Govindan
University of Southern California
Los Angeles, CA, USA

Xiaowei Yang
Duke University
Durham, NC, USA

Abstract

Large cloud providers including AWS, Azure, and Google offer two tiers of network services to their customers: WAN-transit service and inet-transit service. Little is known about how each cloud provider offers 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 the WAN of each of the three clouds. In addition, we measure the access latencies of these two network services of each cloud and compare them with emulated alternative routing strategies.

CCS Concepts

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

Keywords

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. In *Abstracts of the 2025 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS Abstracts '25), June 9–13, 2025, Stony Brook, NY, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3726854.3727321

1 Introduction

Currently, large cloud providers such as AWS, Azure, and Google provide two tiers of network services: an inet-transit service and a WAN-transit service [1–3]. According to the description of cloud providers [1–3], the inet-transit service carries traffic from clients to cloud virtual machines (VMs) over the Internet, while the WAN-transit service uses the cloud's private wide-area network (WAN) to carry the traffic. In theory, since a cloud's private WAN is usually well developed, the WAN-transit service is expected to provide better performance than the inet-transit service.

Technically, cloud providers implement these two services using different BGP prefix announcement strategies. For example, in the inet-transit service, a cloud announces the IP prefix of a VM through a single Point of Presence (PoP) close to the VM. In contrast, the cloud provider announces the prefix through all its PoPs around the world (aka. global anycast) for the WAN-transit service.

In this paper, we investigate three questions: (1) what are cloud providers' strategies of routing these two tiers of services? (2) how does the performance of inet-transit compare with WAN-transit? (3) Is there alternative strategies that can further improve the latency from Internet clients to cloud VMs?

Although prior work by Arnold et al. [5] compared the latency of these two services in Google and AWS, they did not investigate alternative strategies or study Azure's services. In this paper, we conduct a large-scale measurement of the inet-transit and WAN-transit services in AWS, Google, and Azure to thoroughly investigate these three questions. We measure the route and latency information of traffic between more than 5000 RIPE Atlas vantage points [9] and cloud VMs in seven regions in three clouds. Our findings can be summarized as follows:

(1) Cloud providers use global anycast to implement WAN-transit service, but use global anycast, regional anycast, and unicast to implement inet-transit service (§3.1), which is contrary to their service descriptions [1, 3].
(2) Although WAN-transit's ingress traffic often enters a cloud WAN early, the egress traffic sometimes exits the WAN early to reach the users (§3.1).
(3) The path efficiency of inet-transit and WAN-transit services of the three cloud providers are comparable, with WAN-transit being slightly better in some regions and worse in others (§3.2).
(4) We explore the alternative routing strategies that potentially reduce latency. We find that a performance-aware routing strategy can significantly reduce latencies in all three cloud providers for 4% to 85% of vantage point and cloud region pairs (§3.3).

The full version of this paper appears in [8].

2 Methodology

At a high level, we use RIPE Atlas probes and cloud VMs to send and receive repeated traceroute and ping measurements to investigate clouds' tiered network services.

Measurement infrastructures. We use RIPE Atlas probes [9] as the vantage points in our measurements. RIPE Atlas hosted around 12,800 probes around the world at the time of this work. To reduce

*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.

SIGMETRICS Abstracts '25, Stony Brook, NY, USA

ACM ISBN 979-8-4007-1593-8/2025/06
https://doi.org/10.1145/3726854.3727321

the credit consumption of RIPE Atlas, we group the probes by their AS numbers (ASN)s and locations, and we randomly select one probe from a <ASN, metro> group. Finally, we adopt 5205 probes from distinct <ASN, metro> groups for measurements.

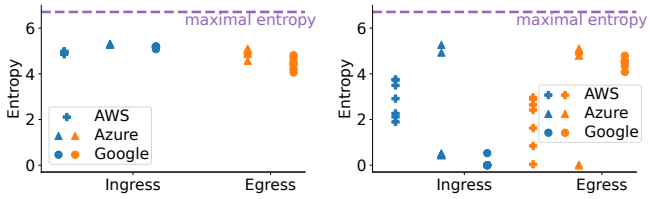
Moreover, we employ VMs in seven regions of three clouds, covering Africa (Cape Town & Johannesburg, AF), Asia (Mumbai & Pune AS), Middle East (Doha & Tel Aviv, ME), Oceania (Sydney, OC), Europe (Frankfurt, EU), North America (Virginia, NA), South America (São Paulo, SA). For each VM, we bind two IP addresses to it for inet-transit and WAN-transit traffic, respectively.

Cloud edge discovery. To investigate the clouds' routing strategies, we detect the ingress and egress cloud edges from traceroutes between selected probes and the clouds. In a traceroute, we first annotate each public IP address with its ASN and geolocation information. We develop accurate IP-to-ASN mapping and IP geolocation techniques in our paper by combining multiple data sources [8]. With such information, in an ingress traceroute to a cloud, we collect the first responsive hop belonging to the cloud ("*cloud-end edge IP*") and its preceding responsive hop ("*neighbor-end edge IP*"). We consider these two hops constitute a valid cloud edge when (a) they are *not* separated by unresponsive hops, or (b) when they are separated by unresponsive hops, but their distance is < 50km (in the same metro) and their RTT difference is less than 2ms.

Alternative route exploration. To investigate the potential latency improvement, we synthesize alternative policy-compliant routes as follows. Our key idea is to choose alternative cloud edges different from the original edge between a probe and a VM. For each *ingress AS* (an AS owning any neighbor-end edge IP), we aggregate all traceroutes to collect all cloud edges (IPs and locations) that peer with this ingress AS. If in a traceroute, we observe that a probe reaches the cloud via an ingress AS, we consider all cloud edges peering with the same ingress AS as the alternative edges for carrying this probe's traffic. Moreover, if a probe reaches the cloud through different ingress ASes in our traceroutes, we aggregate all these ingress ASes' corresponding edges as the probe's alternative edges. This method produces alternative policy-compliant routes because common BGP policies are specified at the AS level [4, 7].

To measure an alternative route's RTT, we send pings from the probe and the VM to the alternative cloud edge. We use the sum of the RTTs of the two segments (*i.e.* a probe-to-edge segment and a VM-to-edge segment) to approximate the RTT of the alternative route. We use the cloud-end edge IP in pings if it is responsive, otherwise we attempt to use the neighbor-end edge IP.

Measurement workflow. To obtain reliable results with acceptable RIPE Atlas credit cost, our measurement consists of three experiments. (1) Firstly, each selected probe sends three to nine pings and one traceroute to our cloud VMs through different tiers of network services and vice versa. (2) With this experiment's data, we obtain the cloud edges and synthesize the alternative routes. Then we use both probes and VMs to ping all alternative edges three to nine times to obtain alternative routes' RTTs as discussed above. (3) Finally, with the preliminary RTT results of alternative routes, we select the routes whose minimum RTTs are lower than the minimum RTT measured by the probe-to-VM pings in the first experiment. We consider these alternative routes provide potential latency improvement, and we measure their RTTs as well as the inet-transit and WAN-transit routes' RTTs by repeating 96 pings over 24 hours to obtain reliable results. We use the median of these 96 RTT samples as the median RTT of an alternative route.



(a) Entropy (WAN-transit) **(b) Entropy (inet-transit)**

Figure 1: The entropy metric that captures the ingress/egress edge diversity of ⟨probe, VM⟩ pairs.

3 Results

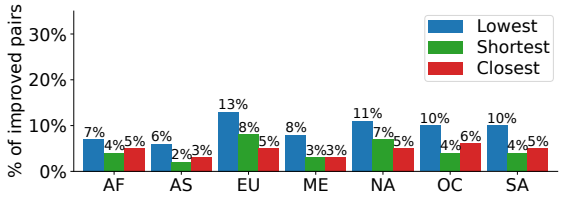
3.1 Routing Strategies

We use the distribution of ingress/egress cloud edges from probes to VMs to infer a cloud provider's WAN-transit and inet-transit services' routing strategies. Specifically, we use entropy to quantify the variety of the ingress/egress edge distributions of a network service. We estimate the probability (p_e) that a ⟨probe, VM⟩ pair ingresses (or egresses) at a cloud edge e as the number of ⟨probe, VM⟩ pairs ingressing (or egressing) at the edge e divided by the total number of ⟨probe, VM⟩ pairs. We compute the entropy 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 entropy indicates more evenly distributed cloud edges. The numbers of cloud edges (by metro locations) uncovered by our method for the three clouds range from 97 to 107. Therefore, the maximum entropy is $log_2 107 \approx 6.7$.

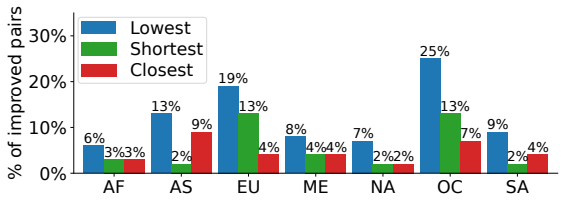
Figure 1 shows the entropy values for two network services. For the WAN-transit service, the entropy values of ingress and egress edges for all clouds are relatively high (> 4), indicating that all three clouds use global anycast for WAN-transit traffic.

However, cloud providers use different routing strategies for their inet-transit services. For AWS's ingress traffic, the entropy values range from 1.9 to 3.8. By manual inspection, we find that AWS uses regional anycast to route the traffic, and thus the edges are distributed in a large region but not globally, leading to a middle range of the entropy. AWS also has diverse strategies for egress edge selection. For Azure, we observe two ingress routing strategies: global anycast and predominantly unicast, indicated by two groups of entropy in the figure. Azure uses two similar strategies for its egress traffic. Moreover, Google consistently uses unicast for its inet-transit ingress traffic, leading to low entropy. However, its egress traffic uses geographically distributed edges.

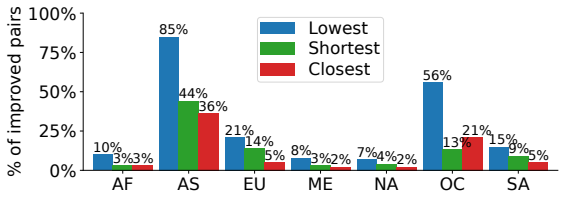
In addition to the entropy, we investigate the extra distance a probe's packet travels compared to entering or leaving a cloud WAN via its closest edge in the WAN-transit service of three clouds. We observe that 59% (AWS), 55% (Azure), and 59% (Google) of the WAN-transit ⟨probe, VM⟩ pairs enter a cloud's WAN at the probes' closest cloud edges. 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. Overall, we find that Azure and Google do not always route their WAN-transit traffic to the cloud edges nearest to the destinations, leading to potential latency inflation.



(a) AWS, $\geq 2500km$, $\geq 10ms$



(b) Azure, $\geq 2500km$, $\geq 10ms$



(c) Google, $\geq 2500km$, $\geq 10ms$

Figure 2: The bars show the percentages of ⟨probe, VM⟩ pairs which have the RTT reduction larger than 10ms. This figure only includes the pairs with a distance between ⟨probe, VM⟩ larger than 2500km. The numbers above bars indicate the concrete percentage. The x-axis shows region abbreviations.

3.2 WAN-Transit Service vs Inet-Transit Service

We compare two network services' performance through a distance-normalized latency metric: the *path inflation factor* [6, 8]. We use this metric to quantify how "fast" a route moves traffic between two endpoints. The larger this number, the slower the traffic travels.

We compute the median value of path inflation factors for different cloud regions and network service tiers. For all investigated clouds and regions, the path inflation factors of a cloud region's WAN-transit service and inet-transit service are close, and the difference of their median values is within 0.3. For every 1000km distance between two locations, every 0.1 increment in this factor corresponds to about a 1ms increment in the RTT. Among 20 combinations of three clouds and seven regions[1], ten of them show that the WAN-transit service is slightly more efficient than the inet-transit service, and in other eight combinations, two network services achieve very close path efficiency (difference of the factor <0.1). Finally, inet-transit service is more efficient than WAN-transit in two combinations (AWS in Europe and Google in Asia).

3.3 Alternative routes

Finally, to explore the performance of alternative routes, we emulate three alternative routing strategies using the methods in §2:

`Lowest-Latency`: We emulate performance-aware routing by synthesizing a route such that the sum of the two segments' RTTs is the lowest among the WAN-transit, the inet-transit, and all synthesized alternative routes.

`Closest-Edge`: We synthesize an alternative route using the closest cloud edge to the probe for both the ingress and egress directions.

`Shortest-Distance`: We synthesize an alternative route such that the sum of the geodesic distances of probe–edge and edge–VM is the shortest among all synthesized alternative routes.

We analyze the ⟨probe, VM⟩ pairs separately based on whether the distance between the probe and the VM is greater or less than 2500km. In this abstract, we present results only for pairs with distances greater than 2500km, using a 10ms threshold to define significant latency improvement. We refer the readers to [8] for results of other ⟨probe, VM⟩ pairs and other improvement thresholds.

Figure 2 shows the alternative route strategies' latency reductions when compared to the WAN-transit service. `Lowest-Latency` achieves the most improvement as it explores alternative routes extensively. In contrast, `Closest-Edge` usually provides the least improvement, while `Shortest-Distance` is in the middle.

For `Lowest-Latency` routing, AWS has 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 pairs' RTTs can be improved by more than 100 ms (See [8]). When taking a closer look, 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 alternative routes use the path through the Eurasia continent, significantly reducing the travel distance by more than 10,000 km.

[1]Google does not have inet-transit service in its Africa region

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. 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] 2024. AWS Global Accelerator Features. https://aws.amazon.com/global-accelerator/features/.
[2] 2024. Network Service Tiers. https://cloud.google.com/network-tiers.
[3] 2024. What is Routing Preference? https://learn.microsoft.com/en-us/azure/virtual-network/ip-services/routing-preference-overview.
[4] 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.
[5] 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.
[6] 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.
[7] Lixin Gao and Jennifer Rexford. 2001. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking* 9, 6 (2001), 681–692.
[8] Shihan Lin, Yi Zhou, Xiao Zhang, Todd Arnold, Ramesh Govindan, and Xiaowei Yang. 2025. Tiered Cloud Routing: Methodology, Latency, and Improvement. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS'25)* 9, 1 (2025), 1–41.
[9] RIPE NCC Staff. 2015. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal* 18 (2015).