Research Proposal

Controllable, Reliable, and Safe Ingress Routing to the Cloud

Jiangchen Zhu
Columbia University

1 INTRODUCTION

Clouds run numerous applications demanding low latency and high availability and serve clients from many geographically distributed *sites*. To meet diverse objectives under fluctuating network conditions - such as reducing latency, load balancing between sites and routes, and ensuring fast failover during failures - the cloud needs complete and timely control over the routes its clients use to reach its sites, i.e., *ingress routing* **TBD: cite**.

Two protocols are crucial for clients to reach the cloud: DNS and BGP. The clients first learn an IP address to access the cloud service (a DNS record) from the DNS server. When accessing the cloud, the packets destined to that address are sent along routes decided by BGP on the Internet. However, Both DNS and BGP have a number of known problems that make ingress routing control challenging.

DNS records are cached by clients' recursive resolvers, applications, and operating systems, which delays DNS updates on the client side. This hurts the cloud's availability when a previously cached IP address becomes unreachable due to failures. Each DNS record carries a time-to-live (TTL) value, determining how long a DNS record should be cached before expiring. Setting a low TTL causes clients to query the DNS server more frequently, which can slow down applications. Moreover, a shorter TTL value does not ensure timely DNS updates because many applications disrespect the TTL and continue using expired DNS records. Our analysis indicates that 13% of client connections are started after the DNS caches have expired, at median beginning 56 seconds after expiration. Specifically for cloud traffic, between 20-85% of traffic occurs more than a minute after the DNS TTL has expired.

BGP decides the path a client takes to access the cloud, but this decision is jointly made by the client network and others on the Internet, each following their own routing policies, thus being out of the cloud's control. BGP was not designed with the cloud's objectives, such as minimizing latency and load balancing, in mind, thus the absence of cloud's control over ingress routes leads to several issues. For instance, BGP may select routes with suboptimal performance, resulting in path inflation [3, 5, 9, 11, 22]. Moreover, load balancing becomes challenging when excessive clients converge on identical routes or target the same site [7, 13]. BGP also suffers from convergence issues: BGP updates cause networks to reselect routes, potentially taking minutes to finalize their decisions, which results in higher packet loss and latency [10, 20]. Furthermore, rapidly updating BGP routing on a global scale contradicts operational best practices, as highlighted in a recent Google study [12]. Such operations are deemed *unsafe*, as any misconfiguration can quickly spread worldwide, triggering cascading failures. This significantly constrains the cloud's ability to safely respond to site failures.

Though these limitations are longstanding, clouds still lack universally applicable and deployable solutions for controlling ingress routing. Systems like Google's Espresso and Facebook's EdgeFabric have been developed to optimize egress routes (from cloud to clients), demonstrating the significance of route selection control for cloud [17, 21]. Controlling egress routes is relatively straightforward since the cloud itself makes the route decisions. In contrast, ingress route control remains challenging because it depends on client networks, which are outside the cloud's control. For ingress routing, two BGP announcement strategies, unicast (with DNS-based redirection) and anycast, are commonly utilized. However, these methods suffer from the limitations in DNS and BGP protocols. Specifically, unicast is hindered by DNS caching, which delays the failover of clients when a site fails. Anycast, on the other hand, compromises the cloud's control over which site clients are directed to, resulting in suboptimal performance and inadequate load balancing.

Some recent work improves ingress routing control but requires collaboration from customer networks or application developers. Systems such as PAINTER and TANGO can only be deployed at collaborative customer networks [4, 9]. Solutions such as application-based redirect and multi-path transport protocols require application support, and their initial connection still uses DNS and BGP so their limitations still exist.

Moreover, studying cloud routing problems poses significant challenges for academic researchers, primarily because they lack the ability to test their routing solutions in real cloud networks on the real Internet. A typical cloud infrastructure spans tens to hundreds of sites worldwide, each connecting to hundreds of peers. While some testbeds enable researchers to conduct BGP routing experiments, their scope and scale fall short of faithfully emulating a cloud network [2, 16]. Conversely, simulating the Internet within a laboratory setting often fails to yield compelling results due to the complexities of accurately replicating both an accurate Internet topology and the networks' intricate routing policies, which are critical yet often undisclosed components of Internet routing **TBD: cite**.

My contributions have played a crucial role in advancing academic research on Internet routing at cloud scale and in the development of practical techniques to improve cloud ingress routing control, which were previously unattainable for cloud networks. These techniques adhere to existing Internet protocols and do not require external collaboration for easy deployment. Instead, they smartly leverage underutilized variables within these protocols that have rarely been considered in the context of cloud ingress routing.

- **Expanding PEERING testbed to cloud scale** (§3.1). I collaborated with Vultr to expand PEERING [16], a BGP routing testbed, to cloud scale, enabling *selective* BGP routing updates from 32 global locations to more than 5000 peers with customizable attributes such as AS path and BGP communities. This expansion makes realistic cloud routing research possible. For instance, a recent SIGCOMM paper leveraged this expanded testbed to develop and assess new ingress routing solutions for clouds [9]. Another study, recently published in NSDI, adopted a similar methodology, though researchers had to coordinate directly with the cloud provider to configure BGP [4]. The expanded PEERING footprint is set to greatly simplify future research efforts in this field.

- **Establishing fundamental tradeoffs in cloud ingress routing and developing techniques that achieve previously unattainable tradeoffs** (§3.2). While the currently employed unicast and anycast techniques fall short of meeting certain objectives due to the limitations of DNS and BGP, it had been unclear whether a fundamental tradeoff is inherent in cloud ingress routing or if an "ideal" technique could exist. I proved an unavoidable tradeoff among control, availability, and operational safety in designing ingress routing solutions, a decision that must be tailored to a cloud's specific business needs. I then developed and evaluated new techniques that combine the strengths of existing methods, pushing them closer to the (unattainable) ideal.
- **Achieving cloud routing objectives with higher ingress route control** (§3.3). I propose to develop new systems that take cloud's ingress routing objectives and network conditions as input to optimize the prefix announcement strategies from its sites. The objectives include minimizing the overall clients latency and load balancing between provider links. This requires the cloud to have rich ingress route control by tailoring where and to whom these announcements are made. Controlling what BGP route a directly neighboring network receives is straightforward, but influencing networks beyond a one-hop distance, where they are free to select from various BGP announcements they have learnt, is challenging. Fortunately, BGP communities allow the cloud to direct how neighboring networks further propagate its announcements, providing a mechanism by which the cloud can potentially further influence the routes of distant clients to increase its ingress control. However, the complexity of BGP communities, with their arbitrary formats and different types of actions documented on network-specific websites, makes manual interpretation and verification time-consuming and uncertain. Automating the learning and verification of BGP communities is a critical first step towards providing more controlled ingress routing options to clients. Future challenges include assessing the benefit of a diverse set of ingress routes in achieving routing objectives and creating a scalable framework that integrates various BGP communities to optimize routing for many client networks.

2 RELATED WORK

Limitations of DNS. One prior work shows how using DNS can control clients to specific sites, achieving ingress site control [6]. Much has discussed the challenges of DNS TTL violation in cloud routing availability [1, 8, 14].

Limitations of BGP.

3 DETAILS OF MY CONTRIBUTIONS

I aim to develop techniques that enable the cloud to meet its ingress routing objectives, such as improved latency and availability. Achieving these goals requires the cloud to have timely and flexible control over how external networks select their routes, a level of control that was previously unattainable without collaboration with them. To aid the research community in studying these routing challenges, I first upgraded a testbed to match the scale of a medium-sized cloud (§3.1). My proposed techniques continue to utilize existing Internet protocols, which allows for straightforward deployment. However, these techniques uniquely leverage underexplored variables within these protocols, enhancing their effectiveness (§3.2, §3.3).

3.1 Expanding the PEERING Testbed

The PEERING testbed allows researchers to make customizable BGP announcements selectively to its neighboring networks, enhancing the study of BGP routing [16]. Despite its utility in numerous innovative studies [9, 15, 18, 19, 22], PEERING's scale, 14 global locations and 5 IXP connections, is limited for routing studies with larger scale (e.g., cloud). Vultr, a cloud provider, enhances this by offering a "bring your own IP" service. This service lets its customers announce their IP spaces and choose which neighboring networks receive these announcements by tagging the relevant BGP communities. It also allows modification of attributes such as AS path and BGP communities in the BGP updates.

After integrating PEERING infrastructure with Vultr servers, researchers can now make BGP announcements from 32 additional locations, reaching over 5000 neighboring networks selectively. The expanded locations span several continents: 11 in North America, 2 in South America, 8 in Asia, 8 in Europe, 2 in Oceania, and 1 in Africa.

This expanded footprint provides a unique platform for addressing cloud routing challenges previously unexplored. A recent SIGCOMM paper utilized this enhanced testbed for cloud-enterprise collaborative routing studies [9]. The expanded capabilities can also retroactively enhance many studies that previously utilized the PEERING testbed [15, 22].

3.2 Establishing Fundamental Tradeoffs in Cloud Ingress Routing and Developing Better Techniques

The clouds have employed two strategies when routing their clients to their sites: unicast with DNS-based redirection and anycast. In unicast, each site announces a unique IP prefix and the DNS server returns clients an address within the prefix of a specific site to redirect the clients to that site. Despite the complete cloud control on clients redirection, the continued use of DNS caching after expiration makes quick DNS updates on client side impossible, delaying the failover when a site fails for minutes and losing availability. On the other hand, in anycast, each site announces the same IP prefix, but the cloud compromises its *control* on which site a client will be routed to since the decision is made by the client networks and other networks instead of the cloud itself. Consequently, clients can end up being routed to geographically distant sites, experiencing increased latency. The lack of control also complicates load balancing between sites.

I define three key *goals* for cloud ingress routing: *control*, *availability* and *safety*. Control refers to the cloud's capability of routing

Technique	Control	Availability	Safety
anycast	low	high	high
unicast	high	low	high
My technique A	high	medium	high
My technique B	high	high	medium
My technique C	high-	high-	high
My technique D	high-	high	high

Table 1: Ingress routing technique tradeoffs among three goals: control, availability and safety. Minus signs are used to indicate being slightly worse in achieving a certain goal.

clients to any site it wants. Availability refers to quickly rerouting clients from a failed site. Safety refers to avoiding updating BGP configurations on healthy sites to minimize the risk of cascading failure as quick global routing reconfiguration contradicts operational best practices. I first observe a fundamental tension: while control benefits from each site announcing a unique IP prefix, guaranteeing any packets destined to the prefix to arrive at the site, availability benefits from multiple sites announcing the same prefix, rerouting clients to healthy sites without waiting for the slow DNS update during site outage. Maximizing both control and availability requires changing BGP announcements upon site failure, compromising safety. I present a more formal proof in a recent submission.

Following this fundamental tradeoff, I develop and evaluate four techniques using the expanded PEERING testbed on the real Internet. Table 1 summarizes all comparisons among my proposed techniques with respect to CDN routing goals. The new techniques achieve previously unattainable combinations of goals, with no existing techniques that can beat them on one goal without compromising on another. Together, they form a new set of "best techniques" for CDN routing. The preliminary work on these techniques (with worse tradeoffs) was published in IMC and awarded a best short paper. The refined techniques are presented in a recently submitted paper.

3.3 Achieving Cloud Routing Objectives with Higher Ingress Route Control

While the techniques in Section 3.2 manage to achieve the control on which *site* clients ingress the cloud without compromising much availability or safety, they have not yet achieved *ingress route* control. I plan to develop systems that take the cloud's routing objective as the input and decide how BGP routes are announced from each site. The objectives can vary from minimizing client latency overall or based on traffic priority, load balancing between links/paths, and facilitating failover after site/link outage. To achieve these objectives, cloud first needs rich control on the ingress route the clients take. Although it is straightforward to control the ingress routing for cloud's neighboring networks by making tailored announcements directly to them, it is hard to achieve this for many more networks that are beyond a one-hop distance since they are free to select any BGP routes learnt from other networks.

Fortunately, the BGP community provides a powerful yet previously unexplored mechanism for influencing the routes of distant client networks. The cloud can direct how its directly neighboring networks further propagate its announcements to distant client networks by tagging the routes with tailored BGP communities. However, BGP communities are 32-bit values with arbitrary formats and network-specific interpretations documented on their websites. For example, two provider networks AS3257 and AS2914 have BGP communities with the same interpretation "do not announce to peers" but with drastically different values 65535:65284 and 2914:429. The types of actions encoded in BGP communities also vary significantly between networks. For example, 65501:nnn is a BGP community of AS2914 which means "prepend to a specific peer nnn 1x", allowing its customers to specify any peer network to apply the action. In contrast, the BGP communities from AS3491 do not support applying to any specific network, but instead have one particular value for each of its peer networks (e.g., 3491:60041 means "prepend to AS174 1x" and there are numerous other values for other networks).

Due to the diversity and a large volume of BGP communities, manual collection and interpretation of BGP communities is time-consuming and unscalable. Automating the learning and verification of BGP communities is the first challenge in realizing a system to achieve ingress route control. I have developed a set of techniques that leverage the latest developments in NLP tools to interpret the semantics of BGP communities and designed tailored Internet measurements to verify their interpretation. Here are the remaining challenges and planned milestones:

To manage clients' ingress routes effectively using BGP communities, the system must first be capable of predicting the varied impacts of specific BGP community actions on clients' ingress routes. These can range widely - for example, some may target a single network while others affect multiple; some have global impacts, while others are regional. It then needs to align these predicted impacts with the objectives, such as minimizing client latency or load balancing between links and sites. The above processes demand extensive measurements to measure or predict route latencies as well as client networks' preferences for different exposable ingress routes. The next step involves searching among a vast set of potential ingress routing options to optimize the announcement strategy for each IP prefix to achieve the desired objectives. A practical method might involve a greedy algorithm, selecting the most beneficial BGP configuration update iteratively and stopping when additional changes yield marginal benefits.

4 CONCLUSION

Existing routing techniques employed by the cloud are insufficient in achieving their routing objectives such as latency, load balancing and reliability, due to the limitations of DNS and BGP. The expansion of the PEERING testbed to cloud scale benefits my study of cloud ingress routing as well as the research community. I then propose and evaluate new techniques that significantly improve cloud's control over clients' ingress routes. My techniques for improving cloud ingress routing achieve previously unattainable tradeoffs and a fine-grained control.

REFERENCES

[1] Mark Allman. Putting DNS in Context. In *ACM IMC*, 2020.

[2] Leandro M. Bertholdo, João M. Ceron, Wouter B. de Vries, Ricardo de Oliveira Schmidt, Lisandro Zambenedetti Granville, Roland van Rijswijk-Deij, and Aiko Pras. TANGLED: A Cooperative Anycast Testbed. In *IFIP/IEEE IM*, 2021.

[3] Henry Birge-Lee, Maria Apostolaki, and Jennifer Rexford. It Takes Two to Tango: Cooperative Edge-to-Edge Routing. In *HOTNETS*, 2022.

[4] Henry Birge-Lee, Sophia Yoo, Benjamin Herber, Jennifer Rexford, and Maria Apostolaki. TANGO: Secure Collaborative Route Control across the Public Internet. In *NSDI*, 2024.

[5] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the Performance of an Anycast CDN. In *ACM IMC*, 2015.

[6] Fangfei Chen, Ramesh K. Sitaraman, and Marcelo Torres. End-User Mapping: Next Generation Request Routing for Content Delivery. In *ACM SIGCOMM*, 2015.

[7] Ashley Flavel, Pradeepkumar Mani, David Maltz, Nick Holt, Jie Liu, Yingying Chen, and Oleg Surmachev. Fastroute: A scalable Load-Aware Anycast Routing Architecture for Modern CDNs. In *USENIX NSDI*, 2015.

[8] Jaeyeon Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. In *ACM SIGCOMM IMW*, 2001.

[9] Thomas Koch, Shuyue Yu, Sharad Agarwal, Ethan Katz-Bassett, and Ryan Beckett. PAINTER: Ingress Traffic Engineering and Routing for Enterprise Cloud Networks. In *ACM SIGCOMM*, 2023.

[10] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence. In *ACM SIGCOMM*, 2000.

[11] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Internet Anycast: Performance, Problems, & Potential. In *ACM SIGCOMM*, 2018.

[12] Bingzhe Liu, Colin Scott, Mukarram Tariq, Andrew Ferguson, Phillipa Gill, Richard Alimi, Omid Alipourfard, Deepak Arulkannan, Virginia Jean Beauregard, Patrick Conner, P. Brighten Godfrey, Xander Lin, Joon Ong, Mayur Patel, Amr Sabaa, Arjun Singh, Alex Smirnov, Manish Verma, Prerepa V Viswanadham, and Amin Vahdat. CAPA: An Architecture For Operating Cluster Networks With High Availability. In *USENIX NSDI*, 2024.

[13] Michael Markovitch, Sharad Agarwal, Rodrigo Fonseca, Ryan Beckett, Chuanji Zhang, Irena Atov, and Somesh Chaturmohta. TIPSY: Predicting Where Traffic Will Ingress a WAN. In *ACM SIGCOMM*, 2022.

[14] Giovane C. M. Moura, John Heidemann, Ricardo de O. Schmidt, and Wes Hardaker. Cache Me If You Can: Effects of DNS Time-to-Live. In *ACM IMC*, 2019.

[15] A S M Rizvi, Leandro Bertholdo, João Ceron, and John Heidemann. Anycast Agility: Network Playbooks to Fight DDoS. In *USENIX Security*, 2022.

[16] Brandon Schlinker, Todd Arnold, Italo Cunha, and Ethan Katz-Bassett. PEERING: Virtualizing BGP at the Edge for Research. In *ACM CoNEXT*, 2019.

[17] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V. Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *ACM SIGCOMM*, 2017.

[18] Satadal Sengupta, Hyojoon Kim, and Jennifer Rexford. Continuous in-network round-trip time monitoring. In *ACM SIGCOMM*, 2022.

[19] Kevin Vermeulen, Ege Gurmericliler, Italo Cunha, Dave Choffnes, and Ethan Katz-Bassett. Internet Scale Reverse Traceroute. In *ACM IMC*, 2022.

[20] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In *ACM SIGCOMM*, 2006.

[21] Kok-Kiong Yap, Murtaza Motiwala, Jeremy Rahe, Steve Padgett, Matthew Holliman, Gary Baldus, Marcus Hines, Taeeun Kim, Ashok Narayanan, Ankur Jain, et al. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In *ACM SIGCOMM*, 2017.

[22] Jiangchen Zhu, Kevin Vermeulen, Italo Cunha, Ethan Katz-Bassett, and Matt Calder. The best of both worlds: high availability CDN routing without compromising control. In *ACM IMC*, 2022.