Research Proposal

Controllable, Reliable, and Safe Ingress Routing to the Cloud

Jiangchen Zhu
Columbia University

1 INTRODUCTION

Cloud providers 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 [1, 3, 5, 7, 14]. Moreover, load balancing becomes challenging when excessive clients converge on identical routes or target the same site [4, 9]. 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 [6, 12]. Furthermore, rapidly updating BGP routing on a global scale contradicts operational best practices, as highlighted in a recent Google study [8]. 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.

Despite these longstanding limitations, 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 [11, 13]. Controlling egress routes is relatively straightforward since the Cloud itself make 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 [2, 5]. 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 replicating a Cloud network. Conversely, simulating the Internet within a laboratory setting often fails to yield compelling results due to the complexities of accurately replicating both the exact network topology and the intricate intra/inter-domain routing policies, which are critical yet often undisclosed components of Internet routing.

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, which have rarely been considered in the context of Cloud ingress routing.

- **Expanding PEERING Testbed to Cloud scale.** I collaborated with Vultr to expand PEERING [10], a BGP routing testbed, to Cloud scale, enabling *selective* BGP routing updates from 30 global locations to about 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 [5]. Another study, recently published in NSDI, adopted a similar methodology, though researchers had to coordinate directly with the cloud provider to configure BGP [2]. The expanded PEERING footprint is set to greatly simplify future research efforts in this field.
- **Fundamental tradeoffs in Cloud ingress routing and a new Pareto Frontier.** 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 demonstrated 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 ideal.
- **Improving ingress routing flexibility with BGP communities.** To control ingress routing, the Cloud decides the propagation of its BGP announcements across the Internet by tailoring where and to whom these announcements are made. Direct adjustments to neighboring networks are straightforward, but influencing networks beyond a one-hop distance, where they are free to select from various BGP announcements they learnt, is challenging. Fortunately, BGP communities allow the Cloud to direct how neighboring networks propagate its announcements, expanding ingress routing options for distant clients. However, the complexity of BGP communities, with their arbitrary formats and meanings 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. Eventually, I aim to develop new systems that take Cloud's ingress routing objectives and network conditions as input to optimize announcement strategies from its sites.

REFERENCES

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

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

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

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.