MEASURING THE EVOLVING INTERNET IN THE CLOUD COMPUTING ERA: INFRASTRUCTURE, CONNECTIVITY, AND PERFORMANCE

by

BAHADOR YEGANEH

A DISSERTATION

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

December 2019

DISSERTATION APPROVAL PAGE

Student: Bahador Yeganeh

Title: Measuring the Evolving Internet in the Cloud Computing Era: Infrastructure, Connectivity, and Performance

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science by:

Prof. Reza Rejaie	Chair
Prof. Ramakrishnan Durairajan	Co-Chair
Prof. Jun Li	Core Member
Prof. Allen Malony	Core Member
Dr. Walter Willinger	Core Member
Prof. David Levin	Institutional Representative

and

Kate Mondloch	Interim Vice Provost and Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded December 2019

DISSERTATION ABSTRACT

Bahador Yeganeh

Doctor of Philosophy

Department of Computer and Information Science

December 2019

Title: Measuring the Evolving Internet in the Cloud Computing Era: Infrastructure, Connectivity, and Performance

The advent of cloud computing as a means of offering virtualized computing and storage resources has radically transformed how modern enterprises run their business and has also fundamentally changed how today's large cloud providers operate. For example, as these large cloud providers offer an increasing number of ever-more bandwidth-hungry cloud services, they end up carrying a significant fraction of today's Internet traffic. In response, they have started to build-out and operate their private backbone networks and have expanded their service infrastructure by establishing a presence in a growing number of colocation facilities at the Internet's edge. As a result, more and more enterprises across the globe can directly connect (i.e. peer) with any of the large cloud providers so that much of the resulting traffic will traverse these providers' private backbones instead of being exchanged over the public Internet. Furthermore, to reap the benefits of the diversity of these cloud providers' service offerings, enterprises are rapidly adopting multi-cloud deployments in conjunction with multi-cloud strategies (i.e., end-to-end connectivity paths between multiple cloud providers).

While prior studies have focused mainly on various topological and performance-related aspects of the Internet as a whole, little to no attention has

been given to how these emerging cloud-based developments impact connectivity and performance in today's cloud traffic-dominated Internet. This dissertation presents the findings of an active measurement study of the cloud ecosystem of today's Internet. In particular, the study explores the connectivity options available to modern enterprises and examines the performance of the cloud traffic that utilizes the corresponding end-to-end paths. The study's main contributions include (i) studying the locality of traffic for major content providers (including cloud providers) from the edge of the network (ii) capturing and characterizing the peering fabric of a major cloud provider, (iii) characterizing the performance of different multi-cloud strategies and associated end-to-end paths, and (iv) designing a cloud measurement platform and decision support framework for the construction of optimal multi-cloud overlays.

This dissertation contains previously published co-authored material.

CURRICULUM VITAE

NAME OF AUTHOR: Bahador Yeganeh

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
Isfahan University of Technology, Isfahan, Iran

DEGREES AWARDED:

Doctor of Philosophy, Computer and Information Science, 2019, University of Oregon
Bachelor of Science, Computer Engineering, 2013, Isfahan University of Technology

AREAS OF SPECIAL INTEREST:

Internet Measurement
Internet Topology
Cloud Computing
Network Overlays

PROFESSIONAL EXPERIENCE:

Graduate Research Assistant, University of Oregon, Eugene, OR, USA, 2013-2019

Software Engineer, PANA Co, Isfahan, Iran, 2010-2013

Summer Intern, InfoProSys, Isfahan, Iran, 2008

GRANTS, AWARDS AND HONORS:

Internet Measurement Conference (IMC) Travel Grant, 2018

Gurdeep Pall Scholarship in Computer & Information Science University of Oregon, 2018

Phillip Seeley Scholarship in Computer & Information Science University of Oregon, 2017

J. Hubbard Scholarship in Computer & Information Science University of Oregon, 2014

PUBLICATIONS:

Bahador, Yeganeh & Ramakrishnan, Durairajan & Reza, Rejaie & Walter, Willinger (2020). Tondbaz: A Measurement-Informed Multi-cloud Overlay Service. *SIGCOMM - In Preparation*

Bahador, Yeganeh & Ramakrishnan, Durairajan & Reza, Rejaie & Walter, Willinger (2020). A First Comparative Characterization of Multi-cloud Connectivity in Today's Internet. *Passive and Active Measurement Conference (PAM) - In Submission*

Bahador, Yeganeh & Ramakrishnan, Durairajan & Reza, Rejaie & Walter, Willinger (2019). How Cloud Traffic Goes Hiding: A Study of Amazon's Peering Fabric. *Internet Measurement Conference (IMC)*

Reza, Motamedi & Bahador, Yeganeh & Reza, Rejaie & Walter, Willinger & Balakrishnan, Chandrasekaran & Bruce, Maggs (2019). On Mapping the Interconnections in Today's Internet. *Transactions on Netowrking (TON)*

Bahador, Yeganeh & Reza, Rejaie & Walter, Willinger (2017). A View From the Edge: A Stub-AS Perspective of Traffic Localization and its Implications. *Network Traffic Measurement and Analysis Conference (TMA)*

ACKNOWLEDGEMENTS

I would like to thank my parents and sister for their sacrifices, endless support throughout all stages of my life and for encouraging me to always pursue my goals and dreams even if it meant that I would be living thousands of miles away from them.

I thank my advisor Prof. Reza Rejaie for providing me with the opportunity to pursue a doctoral degree at UO. I am grateful to him as well as my co-advisor Prof. Ramakrishnan Durairajan and collaborator Dr. Walter Willinger for their guidance and feedback in every step of my PhD, the numerous late nights they stayed awake to help me meet deadlines, and for instilling perseverance in me. Completing this dissertation wouldn't have been possible without each one of you.

I am thankful to my committee members Prof. Jun Li, Prof. Allen Malony, and Prof. David Levin for their valuable input and guidance on shaping the direction of my dissertation and for always being available even on the shortest notice.

Lastly, I am grateful for all the wonderful friendship relations that I have formed through the past years. These friends have been akin to a second family and their support and help through the highs and lows of my life has been invaluable to me.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

The Internet since its inception as a network for interconnecting a handful of academic and military networks has gone through constant evolution throughout the years and has become a large scale distributed network spanning the globe that is intertwined with every aspect of our daily lives. Given its importance, we need to study its health, vulnerability, and connectivity. This is only made possible through constant network measurements. Researchers have conducted measurements in order to gain a better understanding of traffic routing through this network, its connectivity structure as well as its performance. Our interest and ability to conduct network measurements can vary in both scopes with respect to the number or size of networks under study as well as the resolution with regards to focusing on networks as a single unit or paying attention to finer network elements such as routers.

The advent of cloud computing can be considered among the most recent and notable changes in the Internet. Cloud providers (CPs) offer an abundance of compute and storage resources in centralized regions in an on-demand basis. Reachability to these remote resources has been made possible via the Internet. Conversely, this shift in computing paradigm has resulted in cloud providers to become one of the main end-points of traffic within today's Internet. These cloud service offerings have fundamentally changed how business is conducted in all segments of the private and public sectors. This, in turn, has transformed the way these companies connect to major cloud service providers to utilize these services. In particular, many companies prefer to bypass the public Internet and directly connect to major cloud service providers at a close-by colocation (or colo) facility

to experience better performance when using these cloud services. In response to these demands, some of the major colo facilities have started to deploy and operate new switching infrastructure called *cloud exchanges* CoreSite (2018); Demchenko et al. (2013). Importantly, in conjunction with this new infrastructure, these colo providers have also introduced a new interconnection service offering called *"virtual private interconnection (VPI)"* Amazon (2018a); Google (2018a); Microsoft (2018a). By purchasing a single port on the cloud exchange switching fabric in a given facility, VPIs enable enterprises that are either natively deployed in that facility to establish direct peering to any number of cloud service providers that are present on that exchange. Furthermore, there is the emergence of new Internet players in the form of third-party private connectivity providers (e.g. DataPipe, HopOne, among others Amazon (2018c); Google (2018b); Microsoft (2018c)). These entities offer direct, secure, private, layer 3 connectivity between CPs (henceforth referred to as *third-party private* (TPP)) and extend the reach of peering points towards CPs to non-native colo facilities in a wider geographic footprint. TPP routes bypass the public Internet at Cloud Exchanges CoreSite (2018); Demchenko et al. (2013) and offer additional benefits to users (e.g. enterprise networks can connect to CPs without owning an Autonomous System Number, or ASN, or physical infrastructure).

The implications of this transformation for the Internet's interconnection ecosystem have been profound. First, the on-demand nature of VPIs introduces a degree of dynamism into the Internet interconnection fabric that has been missing in the past were setting up traditional interconnections of the public or private peering types took days or weeks. Second, once the growing volume of an enterprise's traffic enters an existing VPI to a cloud provider, it is handled entirely

by that cloud provider's private infrastructure (i.e. the cloud provider's private backbone that interconnects its own datacenters) and completely bypasses the public Internet.

The extensive means of connectivity towards cloud providers coupled with the competing market place of multiple CPs has lead enterprises to adopt a mutli-cloud strategy where instead of considering and consuming compute resources as a utility from a single CP, to better satisfy their specific requirements, enterprise networks can pick-and-choose services from multiple participating CPs (e.g. rent storage from one CP, compute resources from another) and establish end-to-end connectivity between them and their on-premises server(s) at the same or different locations. In the process, they also avoid vendor lock-in, enhance the reliability and performance of the selected services, and can reduce the operational cost of deployments. Indeed, according to an industry report from late 2018 Krishna, Cowley, Singh, and Kesterson-Townes (2018), 85% of the enterprises have already adopted multi-cloud strategies, and that number is expected to rise to 98% by 2021. These disparate resources from various CP regions are connected together either via TPP networks, cloud-providers private (CPP) backbone, or simply via the best-effort public Internet (BEP).

The aforementioned market trends collectively showcase the implications of the cloud computing paradigm on the Internet's structure and topology and highlight the need for focusing on these emergent technologies to have a correct understanding of the Internet's structure and operation.

1.1 Challenges in Topology Discovery & Internet Measurement

The topology of the Internet has been a key enabler for studying routing of traffic in addition to gaining a better understanding of Internet performance

and resiliency. The measurement of Internet in general and capturing Internet topology in specific is challenging due to many factors, namely (i) scale: the vast scale of the Internet as a network spanning the globe limits our abilities to fully capture its structure, (ii) visibility: our view of the Internet is constrained to the perspective that we are able to glean from the limited number of vantage points we are able to look at it, (iii) dynamic: the Internet as an ever-evolving entity is under constant structural change added to this the existence of redundant routes, backup links, and load-balanced paths limits our ability to fully capture the current state of the Internet's topology, (iv) tools: researchers have relied on tools which were originally designed for troubleshooting purposes and the protocol stack of Internet lacks any inherent methods for identifying topology, and (v) intellectual property: many of the participating entities within the Internet lack incentives for sharing or disclosing data pertaining to their internal structure as often these data are key to their competitive edge.

1.2 Dissertation Scope & Contributions

In this dissertation, we study and assess the impact of the wide adoption of CPs on today's Internet traffic and topology. In a broad sense this dissertation can be categorized into four main parts, namely (i) studying the locality of traffic for major content providers (including CPs) from the edge of the network, (ii) presenting methodologies for capturing the topology surrounding cloud providers with a special focus on VPIs that have been under-looked up to this point, (iii) characterizing and evaluating the performance of various connectivity options towards CPs, and (iv) designing and presenting a measurement platform to support the measurement of cloud environments in addition to a decision support

framework for optimal utilization of cloud paths. The following presents an overview of the main contributions of this dissertation.

1.2.1 Locality of Traffic Footprint. Serving user requests from near-by caches or servers has been a powerful technique for localizing Internet traffic with the intent of providing lower delay and higher throughput to end users while also lowering the cost for network operators. This basic concept has led to the deployment of different types of infrastructures of varying degrees of complexity that large CDNs, CPs, ISPs, and content providers operate to localize their user traffic. This work assesses the nature and implications of traffic localization as experienced by end-users at an actual stub-AS. We report on the localization of traffic for the stub-AS UOnet (AS3582), a Research & Education network operated by the University of Oregon. Based on a complete flow-level view of the delivered traffic from the Internet to UOnet, we characterize the stub-AS's *traffic footprint* (i.e. a detailed assessment of the locality of the delivered traffic by all major content providers), examine how effective individual content providers utilize their built-out infrastructures for localizing their delivered traffic to UOnet, and investigate the impact of traffic localization on perceived throughput by end-users served by UOnet. Our empirical findings offer valuable insights into important practical aspects of content delivery to real-world stub-ASes such as UOnet.

1.2.2 Discovery of Cloud Peering Topology. This works main contribution consists of presenting a third-party, cloud-centric measurement study aimed at discovering and characterizing the unique peerings (along with their types) of Amazon, the largest cloud service provider in the US and worldwide. Each peering typically consists of one or multiple (unique) interconnections between Amazon and a neighboring Autonomous System (AS) that are typically established

at different colocation facilities around the globe. Our study only utilizes publicly available information and data (i.e. no Amazon-proprietary data is used) and is therefore also applicable for discovering the peerings of other large cloud providers. We describe our technique for inferring peerings towards Amazon and pay special attention to inferring the VPIs associated with this largest cloud provider. We also present and evaluate a new method for pinning (i.e. geo-locating) each end of the inferred interconnections or peering links. Our study provides a first look at Amazon's peering fabric. In particular, by grouping Amazon's peerings based on their key features, we illustrate the specific role that each group plays in how Amazon peers with other networks. Overall, our analysis of Amazon's peering fabric highlights how (e.g. using virtual and non-BGP peerings) and where (e.g. at which metro) Amazon's cloud traffic "goes hiding"; that is, bypasses the public Internet. In particular, we show that as large cloud providers such as Amazon aggressively pursue new connect locations closer to the Internet's edge, VPIs are an attractive interconnection option as they *(i) create shortcuts between enterprises at the edge of the network and the large cloud providers (i.e. further contributing to the flattening of the Internet) and* (ii) *ensure that cloud-related traffic is primarily carried over the large cloud providers' private backbones (i.e. not exposed to the unpredictability of the best-effort public Internet).*

1.2.3 Cloud Connectivity Performance. This work aims to empirically examine the different types of multi-cloud connectivity options that are available in today's Internet and investigate their performance characteristics using non-proprietary cloud-centric, active measurements. In the process, we are also interested in attributing the observed characteristics to aspects related to connectivity, routing strategy, or the presence of any performance bottlenecks. To

study multi-cloud connectivity from a C2C perspective, we deploy and interconnect VMs hosted within and across two different geographic regions or availability zones (i.e. CA and VA) of three large cloud providers (i.e. Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure) using the TPP, CPP, and BEP option, respectively. Using this experimental setup, we first compare the stability and/or variability in performance across the three connectivity options using metrics such as delay, throughput, and loss rate over time. We find that CPP routes exhibit lower latency and are more stable when compared to BEP and TPP routes. CPP routes also have higher throughput and exhibit less variation compared to the other two options. In our attempt to explain the subpar performance of TPP routes, we find that inconsistencies in performance characteristics are caused by several factors including border routers, queuing delays, and higher loss-rates of TPP routes. Moreover, we attribute the CPP routes' overall superior performance to the fact that each of the CPs has a private optical backbone, there exists rich inter-CP connectivity, and that the CPs' traffic *always* bypasses (i.e. is invisible to) BEP transits.

1.2.4 Optimal Cloud Overlays. This work focuses on the design of a measurement platform for multi-cloud environments aimed at gaining a better understanding of the connectivity and performance characteristics of inter-cloud connectivity paths. We demonstrate the applicability of this platform by deploying it on all available regions of the top three CPs (i.e. Amazon, Microsoft, and Google) and measure the latency among all regions. Furthermore, we capture the traffic cost models of each CP based on publicly published resources. The measured latencies and cost models are utilized by our optimal overlay construction framework that is capable of constructing overlay networks composed of network

paths within the backbone of CP networks. These overlays satisfy the deployment requirements of an enterprise in terms of target regions, and overall traffic budget. Overall our results demonstrate that CP networks are tightly interconnected with each other. Second, multi-cloud paths exhibit higher latency reductions than single cloud paths; e.g., 67% of all paths, 54% of all intra-CP paths, and 74% of all inter-CP paths experience an improvement in their latencies. Third, although traffic costs vary from location to location and across CPs, the costs are not prohibitively high.

1.3 Dissertation Outline

The remainder of this thesis is organized as follows. We provide a background and overview of studies related to topology discovery and performance characteristics of Internet routes in Chapter II. Next, in Chapter III we characterize the locality of Internet traffic from an edge perspective and demonstrate that the majority of Internet traffic can be attributed to CDNs and cloud providers. Chapter IV presents our work on the discovery of Amazon's peering ecosystem with a special focus on VPIs. We evaluate and characterize the performance of different connectivity options in a multi-cloud setting within Chapter V. Chapter VI presents our proposed measurement platform for multi-cloud environments and showcases the applicability of this measurement platform in the creation of optimal overlays. We conclude and summarize our contributions in Chapter VII.

1.3.1 Navigating the Chapters.

This dissertation studies the effects of cloud-providers on the Internet from multiple perspectives, including (i) traffic, (ii) topology (iii), performance, and (iv) multi-cloud deployments. The chapters presented in this dissertation can be read independently. A reader interested in

Table 1. Topics covered in each chapter of the dissertation.

Chapter	**Traffic**	**Topology**	**Performance**	**Multi-Cloud**
3	X		X	
4		X		
5		X	X	X
6		X	X	X

individual topics can refer to Table 1 for a summary of topics that are covered in each chapter.

CHAPTER II

RELATED WORK

This chapter presents a collection of prior studies for various aspects of Internet measurement to gain insight into the topology of the Internet as well as its implications in designing applications. For Internet measurement, we focus on recent studies regarding the simulation and characterization of Internet topology. Furthermore, we organize these studies based on the resolution of the uncovered topology with an emphasis on the utilized datasets and employed methodologies. On the second part, we focus on various implications of Internet topology on the design and performance of applications. These studies are organized in accordance with the implication of topology on performance or resiliency of the Internet. Furthermore we emphasis on how various resolutions of Internet topology allow researchers to conduct different studies. The collection of these studies present a handful of open and interesting problems regarding the future of Internet topology with the advent of cloud providers and their centrality within today's Internet.

The rest of this chapter is organized as follows. First, in Section 2.1 we present a primer on the Internet and introduce the reader with a few taxonomies that are frequently used within this document. Second, an overview of most common datasets, platforms, and tools which are used for topology discovery is given in Section 2.2. Third, the review for recent studies on Internet topology discovery is presented in Section 2.3. Lastly, Section 2.4 covers the recent studies which utilize Internet topologies to study the performance and resiliency of the Internet.

2.1 Background

The Internet is a globally federated network composed of many networks each of which has complete autonomy over the structure and operation of its own network. These autonomous systems or networks (AS) can be considered as the building blocks of the Internet. Each AS represents a virtual entity and can be composed of a vast network infrastructure composed of networking equipment like routers and switches as well as transit mediums such as Ethernet and fiber optic cables. These ASes can serve various purposes such as providing transit or connectivity for other networks, generating or offering content such as video streams, or merely represent the network of an enterprise. Each of the connectivity provider ASes can be categorized into multiple tiers based on their size and how they are interconnected with other ASes. These tiers create a natural hierarchy of connectivity that is broadly composed of 3 tiers namely, (i) Tier-1: an AS that can reach all other networks without the need to pay for its traffic exchanges, (ii) Tier-2: an AS which can have some transit-free relations with other ASes while still needing to pay for transit for reachability to some portion of the Internet, and (iii) Tier-3: an AS that solely purchases transit for connectivity to the Internet. While each network has full control over its own internal network and can deliver data from one internal node to another, transmitting data from one AS to another requires awareness of a path that can reach the destination AS. This problem is solved by having each AS advertise its own address space to neighboring ASes through the border gateway protocol (BGP). Upon receiving a BGP announcement, each AS would prepend its own AS number (ASN) to the AS-path attribute of this announcement and advertise this message to its own neighbors. This procedure allows ASes to learn about other networks and the set of AS-paths or routes that

they can be reached through. ASes can interconnect with each by linking their border routers at one or multiple physical locations. These border routers are responsible for advertising their prefixes in addition to performing the actual routing of traffic within the Internet. The border routers of ASes are placed within colocation facilities (colo) that offer space, power, security, and networking equipment to the tenants ASes. Each AS can have a physical presence in multiple metro areas. The collection of their routers within each of these metro areas are referred to as the points of presence (PoP) for these ASes. Figure 1 presents a high level abstraction of the aforementioned concepts. The figure consists of 3 ASes namely, AS_A, AS_B, and AS_C in red, blue, and green accordingly. The internal structure ASes is abstracted out presenting only the border routers of each AS. AS_A and AS_B have two PoPs one in LA and another in NY while AS_C is only present in NY. AS_A and AS_B establish a private interconnection with each other through their LA PoP within $colo_1$ while they peer with each other as well as AS_C in their NY PoP in $colo_2$ through an IXPs switching fabric.

2.2 Tools & Datasets

This section provides an overview of various tools and datasets that have been commonly used by the measurement community for discovering Internet topology. We aim to familiarize the reader with these tools and datasets as they are continuously used within the literature by researchers. Researchers have utilized a wide range of tools for the discovery of topologies; they range from generic network troubleshooting tools such as traceroute or paris-traceroute to tools developed by the Internet measurement community such as Sibyl or MIDAR. Furthermore, researchers have benefited from many measurement platforms such as RIPE Atlas

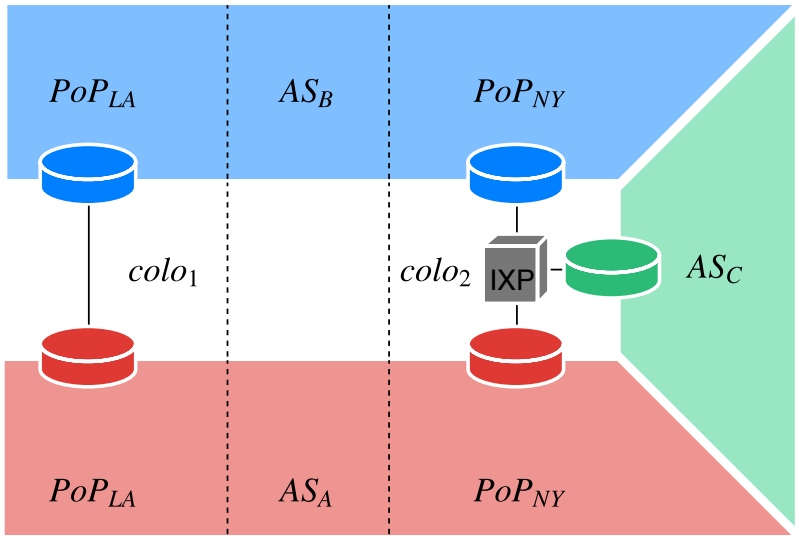


Figure 1. Abstract representation for topology of AS_A, AS_B, and AS_C in red, blue, and green accordingly. AS_A and AS_B establish a private interconnection inside $colo_1$ at their LA PoP while peering with each other as well as AS_C inside $colo_2$ at their NY PoP facilitated by an IXP's switching fabric.

or PlanetLab which enable them to perform their measurements from a diverse set of ASes and geographic locations.

In addition to the aforementioned toolsets researchers have benefited from various datasets within their work. These datasets are collected by a few well-known projects in the Internet measurement community such as Routeviews University of Oregon (2018), CAIDA's Ark CAIDA (2018), and CAIDA's AS relationships datasets or stem from other sources such as IP to geolocation datasets or information readily available on colocation facilities or IXP operators websites.

The remainder of this section is organized within two subsections. First, §2.2.1 would provide an overview of the most commonly used tools and platforms for Internet topology discovery. Second, §2.2.2 would give a brief overview of the datasets that appear in the literature presented within §2.3 and §2.4.

2.2.1 Measurement Tools & Platforms.

Broadly speaking the tools used for Internet topology discovery can be categorized within three groups namely, (i) path discovery, (ii) alias resolution, and (iii) interface name decoding.

2.2.1.1 Path Discovery.

Although originally developed for troubleshooting purposes, *traceroute* Jacobson (1989) has become one of the prominent tools used within the Internet measurement community. *traceroute* displays the set of intermediate router interfaces that are traversed towards a specific destination in the forward path. This is made possible by sending packets towards the destination with incremental TTL values, each router along the path would decrease the TTL value before forwarding the packet. If a router encounters a packet with a TTL value of 0 the packet would be dropped, and a notification message with its source address would be sent back to the originator of the packet. This, in turn, allows the originator of these packets to identify the source address of router interfaces along the forward path. Deployment of load-balancing mechanics by routers which rely on packet header fields can lead to inaccurate and incomplete paths to be reported by *traceroute*. Figure 2 illustrates an example of incorrect inferences by *traceroute* in the presence of load-balanced paths. Node a is a load-balancer and multiplexes packets between the top and bottom paths. In this example, the $TTL = 2$ probe originated from the source traverses the top path and expires at node b while the $TTL = 3$ probe goes through the bottom path and terminates at node e. These successive probes cause *traceroute* to incorrectly infer a non-existent link between nodes b and e. To address this problem, Augustin, Friedman, and Teixeira (2007) developed *paris-traceroute* which relies on packet header contents to enforce load-balancers to pick a single route for all probes of a

single traceroute session. Furthermore, *paris-traceroute* uses a stochastic probing algorithm in order to enumerate all possible interfaces and links at each hop.

Given the scale of the Internet and its geographic span relying on a single vantage point (VP) to conduct topology discovery studies would likely lead to incomplete or inaccurate inferences. Researchers have relied on various active measurement platforms which either host a pre-defined set of tools, e.g., Dasu, Bismark, Dimes, Periscope, and RIPE Atlas Giotsas, Dhamdhere, and Claffy (2016); RIPE NCC (2016); Sánchez et al. (2013); Shavitt and Shir (2005); Sundaresan, Burnett, Feamster, and De Donato (2014) or provide full-access control, e.g. PlanetLab, CAIDA Archipelago, and GENI Berman et al. (2014); Chun et al. (2003); Hyun (2006) to the user to conduct their measurements from a diverse set of networks and geographic locations. For example, RIPE Atlas RIPE NCC (2016) is composed of many small measurement devices (10k at the time of this survey) that are voluntarily hosted within many networks on a global scale. Hosting RIPE Atlas nodes would give credit to the hosting entity which later on could be used to conduct latency (*ping*) and reachability (*traceroute* and *paris-traceroute*) measurements. Periscope Giotsas et al. (2016) is another platform that provides a unified interface for probing around 1.7k publicly available looking glasses (LGs) which provide a web interface to conduct basic network commands (*ping, traceroute,* and *bgp* on routers hosted in roughly 0.3k ASes. Periscope VPs are located at core ASes while RIPE Atlas probes are hosted in a mix of core and edge networks. Dasu Sánchez et al. (2013) on the other hand mainly consists of VPs at edge networks and more specifically broadband users relying on ISPs to have Internet connectivity. Dasu consists of a plugin for the Vuze BitTorrent client that is able to conduct network measurement from the computers of users who

have installed their plugin on their Vuze client. The authors of Dasu incentivize its adoption by reporting broadband network characteristics to its users. Cunha et al. (2016) developed a route oracle platform named Sibyl which allowed users to define the path requirements for their measurement through an expressive input language based on symbolic regular-expressions after which Sibyl would select the source (LG) and destination pair that has the highest likelihood of satisfying the users path requirements based on its internal model.

Lastly, considering the large number of Internet hosts and networks, researchers have developed a series of tools that allow them to conduct large scale measurements in parallel. The methodology of *paris-traceroute* has been incorporated in *scamper* Luckie (2010), an extensible packet prober that implements various common network measurement functionalities such as traceroute, ping, and alias resolution into a single tool. *scamper* is able to conduct measurements in parallel without exceeding a predefined probing rate. While *scamper* is able to run measurements in parallel, each measurement is conducted sequentially, this in turn could hinder its rate or induce overhead to the probing device in order to maintain the state of each measurement. *yarrp* Beverly (2016); Beverly, Durairajan, Plonka, and Rohrer (2018) is a high-rate IPv4 and IPv6 capable, Internet-scale probing tool inspired by the state-less design principles of *ZMap* Durumeric, Wustrow, and Halderman (2013) and *masscan* Graham, Mcmillan, and Tentler (2014). *yarrp* randomly permutates the IP and TTL space and encodes the state information of each probe within the IP and TCP header fields (which are included in the ICMP response) and is therefore able to conduct traceroute probes in parallel without incrementally increasing the TTL value.

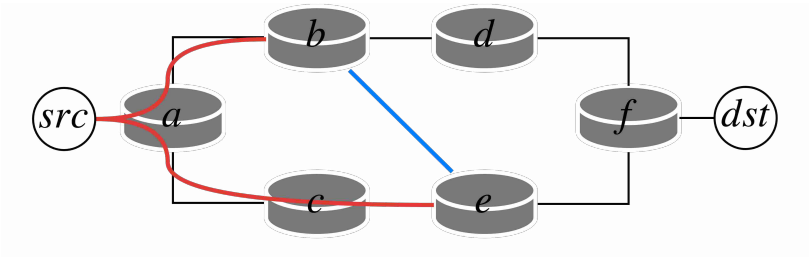


Figure 2. Illustration of inferring and incorrect link $(b-e)$ by *traceroute* due to load balanced paths. Physical links and traversed paths are shown with black and red lines accrodingly. The $TTL = 2$ probe traverses the top path and expires at node b while the $TTL = 3$ probe traverses the bottom path and expires at node e. This succession of probes causes *traceroute* to infer a non-existent link $(b-e)$.

2.2.1.2 Alias Resolution. Paths which are obtained via the tools outlined in §2.2.1.1 all specify the router interfaces that are encountered along the forward path. It is possible to observe multiple interfaces of a single router within different traceroute paths. The association of these interfaces to a single physical router is not clear from these outputs. Alias resolution tools have been developed to solve this issue. These tools would accept a set of interface addresses as an input and would provide a collection of interface sets, each of which corresponds to a single router. Alias resolution tools can broadly be categorized into two groups namely, (i) probing Bender, Sherwood, and Spring (2008); Govindan and Tangmunarunkit (2000); Keys, Hyun, Luckie, and Claffy (2013); Spring, Mahajan, and Wetherall (2002); Tozal and Sarac (2011) and (ii) inference M. Gunes and Sarac (2009); M. H. Gunes and Sarac (2006); Sherwood, Bender, and Spring (2008); Spring, Dontcheva, Rodrig, and Wetherall (2004) based techniques. The former would require a VP which would probe the interfaces in question to identify sets of interfaces which belong to the same router. Probe based techniques mostly rely on the IP ID field which is used for reassembling fragmented packets at the network layer. These techniques assume that routers rely

on a single central incremental counter which assigns these ID values regardless of the interface. Given this assumption, Ally Spring et al. (2002) probed IPs with UDP packets having high port numbers (most likely not in use) to induce an ICMP port unreachable response. Ally will infer IP addresses to be aliases if successive probes have incremental ID values within a short distance. Radargun Bender et al. (2008) tries to address the probing complexity of Ally ($O(n^2)$) by iteratively probing IPs and inferring aliases based on the velocity of IP ID increments for each IP. MIDAR Keys et al. (2013) presents a precise methodology for probing large scale pool of IP addresses by eliminating unlikely IP aliases using a velocity test. Furthermore, aliases are inferred by comparing the monotonicity of IP ID time series for multiple target IP addresses. MIDAR utilizes ICMP, TCP, and UDP probes to increase the likelihood of receiving responses from each router/interface. Palmtree Tozal and Sarac (2011) probes /30 or /31 mates of target IPs using a TTL value inferred to expire at the router in question to induce an ICMP_TTL_EXPIRED response from another interface of the router. Assuming no path changes have happened between measuring the routers hop distance and the time the ICMP_TTL_EXPIRED message has been generated, the source address of the ICMP_TTL_EXPIRED message should reside on the same router of the target IP and therefore are inferred to be aliases.

Inference based techniques accept a series of traceroute outputs and rely on a set of constraints and assumptions regarding the setting and environment which these routers are deployed to make inferences about interfaces that are most likely part of the same router. Spring et al. suggest a common successor heuristic to attribute IP addresses on the prior hop to the same router. This heuristic assumes that no layer-2 devices are present between the two routers in question.

Analytical Alias Resolution (AAR) M. H. Gunes and Sarac (2006) infers aliases using symmetric traceroute pairs by pairing interface addresses using the common address sharing convention of utilizing a /30 or /31 prefix for interfaces on both ends of a physical link. This method requires the routes between both end-pairs to be symmetrical. DisCarte Sherwood et al. (2008) relies on the route record option to capture the forward and reverse interfaces for the first nine hops of a traceroute. Limited support and various route record implementations by routers in addition to the high complexity of the inference algorithm limits its applicability to wide/large scenarios.

2.2.1.3 Interface Name Decoding.

Reverse DNS (RDNS) entries for observed interface addresses can be the source of information for Internet topology researchers. Port type, port speed, geolocation, interconnecting AS, and IXP name are examples of information which can be decoded from RDNS entries of router interfaces. These information sets are embedded by network operators within RDNS entries for ease of management in accordance to a (mostly) structured convention. For example, *ae-4.amazon.atlnga05.us.bb.gin.ntt.net* is an RNDS entry for a router interface residing on the border router of NTT (ntt.net) within Atlanta GA (atlnga) interconnecting with Amazon. Embedding this information is completely optional, and the structure of this information varies from one AS to another. Several tools have been developed to parse and extract the embedded information within RDNS entries Chabarek and Barford (2013); Huffaker, Fomenkov, et al. (2014); Scheitle, Gasser, Sattler, and Carle (2017); Spring et al. (2002). Spring et al. extracted DNS encoded information for the ISPs under study in their *Rocketfuel* project Spring et al. (2002). As part of this process, they relied on the city code names compiled in Padmanabhan and Subramanian

(2001) to search for domain names which encode geoinformation in their name. *PathAudit* Chabarek and Barford (2013) is an extension to traceroute which report encoded information within observed router hops. In addition to geo information, *PathAudit* reports on interface type, port speed, and manufacturing vendor of the router. The authors of *PathAudit* extract common encodings (tags) from device configuration parameters, operator observations, and common naming conventions. Using this set of tags, RDNS entries from CAIDA's Ark project CAIDA (2018) are parsed to match against one or multiple of these tags. A clustering algorithm is employed to identify similar naming structures within domains of a common top level domain TLD. These common structures are translated into parsing rules which can match against other RDNS entries. *DDeC* Huffaker, Fomenkov, and claffy (2014) is a web service which decodes embedded information within RDNS entries by unifying the rulesets obtained by both *UNDNS* Spring et al. (2002) and *DRoP* Huffaker, Fomenkov, et al. (2014) projects.

2.2.2 Datasets. Internet topology studies have been made possible through various data sources regarding BGP routes, IXP information, colo facility listings, AS attributes, and IP to geolocation mapping. The following sub-section provides a short overview of data sources most commonly used by the Internet topology community.

2.2.2.1 BGP Feeds & Route Policies. University of Oregon's RouteViews and RIPE Routing Information Service (RIS) RIPE (2018); University of Oregon (2018) are projects originally conceived to provide real-time information about the global routing system from the standpoint of several route feed collectors. These route collectors periodically report the set of BGP feeds that they receive back to a server where the information is made publicly accessible. The data from

these collectors have been utilized by researchers to map prefixes to their origin-AS or to infer AS relationships based on the set of observed AS-paths from all the route collectors. Routeviews and RIPE RIS provide a window into the global routing system from higher tier networks. Packet Clearing House (PCH) Packet Clearing House (2018) maintains more than 100 route collectors which are placed within IXPs around the globe and provides a complementary view to the global routing system presented by Routeviews and RIPE RIS. Lastly, Regional Internet Registries (RIRs) maintain databases regarding route policies of ASes for each of the prefixes that are delegated to them using the Route Policy Specification Language (RPSL). Historically, RPSL entries are not well adopted and typically are not maintained/updated by ASes. The entries are heavily concentrated within RIPE and ARIN regions but nonetheless have been leveraged by researchers to infer or validate AS relationships Giotsas, Luckie, Huffaker, and Claffy (2015); Giotsas, Luckie, Huffaker, et al. (2014).

2.2.2.2 ***Colocation Facility Information.*** Colocation facilities (colo for short) are data-centers which provide space, power, cooling, security, and network equipment for other ASes to host their servers and also establish interconnections with other ASes that have a presence within the colo. PeeringDB and PCH Packet Clearing House (2017); PeeringDB (2017) maintain information regarding the list of colo facilities and their physical location as well as tenant ASes within each colo. Furthermore, some colo facility operators provide a list of tenant members as well as the list of transit networks that are available for peering within their facilities for marketing purposes on their website. This information has been mainly leveraged by researchers to define a set of constraints regarding the points of presence (PoP) for ASes.

2.2.2.3 IXP Information.

IXPs are central hubs providing rich connectivity opportunities to the participating ASes. Their impact and importance regarding the topology of the Internet have been highlighted within many works Augustin, Krishnamurthy, and Willinger (2009); Castro, Cardona, Gorinsky, and Francois (2014); Comarela, Terzi, and Crovella (2016); Nomikos et al. (2018). IXPs provide a switching fabric within one or many colo facilities where each participating AS connects their border router to this switch to establish bi-lateral peering with other member ASes or establishes a one to many (multi-lateral) peering with the route server that is maintained by the IXP operator. IXP members share a common subnet owned by the IXP operator. Information regarding the location, participating members, and prefixes of IXPs is readily available through PeeringDB, PCH, and the IXP operators website Packet Clearing House (2017); PeeringDB (2017).

2.2.2.4 IP Geolocation.

The physical location of IP addresses isn't known. Additionally, IP addresses could correspond to mobile end-hosts or can be repurposed by the owner AS and therefore have a new geolocation. Several free and commercial databases have been made throughout the years that attempt to map IP addresses to physical locations. These datasets can vary in their coverage as well as the resolution of mapped addresses (country, state, city, and geo-coordinates). Maxmind's GeoIP2 MaxMind (2018), IP2Location databases IP2Location (2018), and NetAcuity NetAcuity (2018) are among the most widely used IP geolocating datasets used by the Internet measurement community. Majority of these datasets have been designed to geolocate end-host IP addresses. Gharaibeh et al. (2017) compare the accuracy of these datasets for geolocating router interfaces and while NetAcuity has relatively higher accuracy than Maxmind and IP2Location

datasets, relying on RTT validated geocoding of RDNS entries is more reliable for geolocating router and core addresses.

2.3 Capturing Network Topology

This section provides an overview of Internet measurement studies which attempt to capture the Internet's topology using various methodologies motivated by different end goals. Capturing Internet topology has been the focus of many pieces of research over the past decade, while each study has made strides of incremental improvements to present a more complete and accurate picture of Internet topology, the problem remains widely open and the subject of many recent studies.

Internet topology discovery has been motivated by a myriad of applications ranging from protocol design, performance measurement in terms of inter-AS congestion, estimating resiliency towards natural disasters and service or network interruptions, security implications of DDoS attacks and much more. A motivating example would be the Netflix Verizon dispute where the subpar performance of Netflix videos for Verizon customers lead to lengthy accusations from both parties Engebretson (2014). The lack of proper methodologies to capture inter-AS congestion by independent entities at the time further elongated the dispute. Within Section 2.4 we provide a complete overview of works which rely on some aspect of Internet topology to drive their research and provide insight regarding the performance or resiliency of the Internet.

Capturing Internet topology is hard due to many contributing factors, the following is a summary of them:

- The Internet is by nature a decentralized entity composed of a network of networks, each of the constituent networks lacks any incentive to share

their topology publicly and often can have financial gains by obscuring this information.

- Topology discovery studies are often based on “hackish" techniques that rely on toolsets which were designed for completely different purposes. The designers of the TCP/IP protocol stack did not envision the problem of topology discovery within their design most likely due to the centralized nature of the Internet in its inception. The *de facto* tool for topology discovery has been *traceroute* which is designed for troubleshooting and displaying paths between a host and a specific target address.

- Capturing inter-AS links within Internet topology becomes even more challenging due to lack of standardization for proper ways to establish these links. More specifically, the shared address between two border routers could originate from either of the participating networks. Although networks typically rely on common good practices such as *using addresses from the upstream provider*, the lack of any oversight or requirement within RFC standards does not guarantee its proper execution within the Internet.

- A certain set of RFCs regarding how routers should handle TTL expired messages has resulted in incorrect inferences of the networks which are establishing inter-AS interconnections. For example, responses generated by third-party interfaces on border routers could lead to the inference of an inter-AS link between networks which necessarily are not interconnected with each other.

Topology discovery studies can be organized according to many of their features; in particular, the granularity of the obtained topology seems to be the

most natural fit. Each of the studies in this section based on the utilized dataset, or devised methodology results in topologies which capture the state of the Internet at different granularities, namely physical-level, router-level, PoP-level, and AS-level. The aforementioned resolutions of topology have a direct mapping to the abstract layers of the TCP/IP stack, e.g. physical-level corresponds to the first layer (physical), router-level can be mapped to the transport layer, and PoP-level as well as AS-level topologies are related to application layer at the top of the TCP/IP stack. These abstractions allow one to capture different features of interest without the need for dealing with the complexities of lower layers. For instance, the interplay of routing and the business relationships between different ASes can be captured through an AS-level topology without the need to understand how and where these inter-AS relationships are being established.

In the following subsections, we will provide an overview of the most recent as well as prominent works that have captured Internet topology at various granularities. We present all studies in accordance to their chronological order starting with works related to AS-level topologies as the most abstract representation of Internet topology within Section §2.3.1, AS-level topologies are the oldest form of Internet topology but have retained their applicability for various forms of analyses throughout the years. Later we'll present router-level and physical-level topologies within Section §2.3.2 and §2.3.4 accordingly.

2.3.1 AS-Level Topology. The Internet is composed of various networks or ASes operating autonomously within their domain that interconnect with each other at various locations. This high-level abstraction of the Internet's structure is captured by graphs representing AS-level topologies where each node is an AS and edges present an interconnection between two ASes. These

graphs lay-out virtual entities (ASes) that are interconnecting with each other and abstract out details such as the number and location where these inter-AS links are established. For example, two large Tier-1 networks such as Level3 and AT&T can establish many inter-AS links through their border routers at various metro areas. These details are abstracted out, and all of these inter-AS links are represented by a single edge within the AS-level topology. The majority of studies rely on control plane data that is obtained by active measurements of retrieving router dumps through available looking glasses or passive measurements that capture BGP feeds, RPSL entries and BGP community attributes. Path measurements captured through active or passive *traceroute* probes have been an additional source of information for obtaining AS-level topologies. The obtained *traceroute* paths have been mapped to their corresponding AS path by translating each hop's address to its corresponding AS. Capturing AS-level topology has been challenging mainly due to limited visibility into the global routing system, more specifically the limited set of BGP feeds that each route collector is able to observe. This limited visibility is known as the topology incompleteness problem within the community. Researchers have attempted to address this issue by either modeling Internet topology by combing the limited ground truth information with a set of constraints or by presenting novel methodologies that merge various data sources in order to obtain a comprehensive view of Internet topology. The later efforts lead to research's that highlighted the importance of IXPs as central hubs of rich connectivity. Within the remainder of this Section we organize works into the following three groups: (i) graph generative and modeling, (ii) topology incompleteness, and (iii) IXP's internal operation and peerings.

2.3.1.1 Graph Generation & Modeling.

Graph generation techniques attempt to simulate network topologies by relying on a set of constraints such as the maximum number of physical ports on a router. These constraints coupled with the limited ground truth information regarding the structure of networks are used to model and generate topologies. The output of these models can be used in other studies which investigate the effects of topology on network performance and resiliency of networks towards attacks or failures caused by natural disasters.

Li, Alderson, Willinger, and Doyle (2004) argue that graph generating models rely on replicating too abstract measures such as degree distribution which are not able to express the complexities/realities of Internet topology. Authors aim to model ASes/ISPs as the building blocks of the Internet at the granularity of routers, where nodes represent routers and links are Layer2 physical links which connect them together. Furthermore, the authors argue that technological constraints on routers switching fabric dictate the amount of bandwidth-links we can have within this topology. Furthermore, due to economical reasons access providers aggregate their traffic over a few links as possible since the cost of laying physical links could surpass that of the switching/routing infrastructure. This, in turn, leads to lower degree core and high degree edge elements. The authors create five graphs with the same degree distribution but based on different heuristics/models and compare the performance of these models using a single router model. Interestingly graphs that are less likely to be produced using statistical measures have the highest performance.

Gregori, Improta, Lenzini, and Orsini (2011) conduct a structural interpretation of the Internet connectivity graph with an AS granularity. They

report on the structural properties of this graph using k-core decomposition techniques. Furthermore, they report what effects IXPs have on the AS-level topology.

The data for this study is compiled from various datasets, namely CAIDA's Ark, DIMES, and Internet Topology Collection from IRL which is a combination of BGP updates from Routeviews, RIPE RIS, and Abilene. The first two datasets consist of traceroute data and are converted to AS-level topologies by mapping each hop to its corresponding ASN. A list of IXPs was obtained using from PCH, PeeringDB, Euro-IX, and bgp4.as. The list of IXP members was compiled either from the IXP websites or by utilizing the **show ip bgp summary command** from IXPs which host an LG.

Using the obtained AS-level graph resulted from combing various data sources the authors report on various characteristics of the graph namely: degree, average neighbor degree, clustering coefficient, betweenness centrality, and k-core decomposition. A k-core subgraph has a minimum degree of k for every node and is the largest subgraph which has this property. The authors present stats regarding the penetration of IXPs in different continents with Europe having the largest share (47%) and North America (19%) at second position. Furthermore using k-core decomposition, the authors identify a densely connected core and a loosely connected periphery which consists of the majority of nodes. The authors also look at the fraction of nodes in the core which are IXP participants and find that IXPs play a fundamental role in the formation of these cores.

*2.3.1.2 **Topology Incompleteness.*** Given the limited visibility of each of the prior works, researchers have relied on a diverse set of data sources and devised new methodologies for inferring additional peerings to address the

incompleteness of Internet topology. These works have lead to highlighting the importance of IXPs as a means of providing the opportunity for establishing many interconnections with IXP members and a major source for identifying missing peering links. Peerings within IXPs and their rich connectivity fabric between many edge networks caused topological changes to the structure of the Internet deviating from the historical hierarchical structure and as a consequence creating a more flat Internet structure referred to as Internet flattening within the literature.

He, Siganos, Faloutsos, and Krishnamurthy (2009) address AS-level topology incompleteness by presenting tools and methodologies which identify and validate missing links. BGP snapshots from various (34 in total) Routeviews, RIPE RIS, and public route servers are collected to create a baseline AS-level topology graph. The business relationship of each AS edge is identified by using the PTE algorithm Xia and Gao (2004). The authors find that the majority of AS links are of a c2p type, while most of the additional links which are found by additional collectors are p2p links. Furthermore, by parsing IRR datasets using Nemecis Siganos and Faloutsos (2004) to infer additional AS links. A list of IXP participants is compiled by gathering IXP prefixes from PCH and performing DNS lookups and parsing the resulting domain name to infer the participating ASN. Furthermore, the authors infer inter-AS links within IXPs by relying on traceroute measurements which cross IXP addresses and utilize a majority voting scheme to infer the participants ASN reliably. By Combing all these datasets and proposed methodologies, the authors find about 300% additional links compared to prior studies, most of which is found to be established through IXPs.

Augustin et al. (2009) attempt to expand on prior works for discovering IXP peering relationships by providing a more comprehensive view of this ecosystem.

They rely on various data sources to gather information on IXPs as much as possible, their data-sources are: (i) IXP databases such as PCH and PeeringDB, (ii) IXP websites which typically list their tenants as well as the prefixes which are employed by them, (iii) RIRs may include BGP policy entries specifically the *import* and *export* entries that expose peering relationships, (iv) DNS names of IXP addresses which include information about the peer, (v) BGP dumps from LGs, Routeviews, and RIPE's RIS can include next hop neighbors which are part of an IXP prefix. The authors conduct targeted traceroute measurements with the intention of revealing peering relationships between members of each IXP. To limit the number of conducted probes, the authors either select a vantage point within one of the member ASes or if not available they rely on the AS relationship datasets to discover a - at most 2 hops away - neighbor for each member which has a VP. Using the selected VPs, they conduct traceroutes towards alive addresses (or random address if such an address was not discovered) in the target network. Inference of peerings based on traceroutes is done using a majority voting scheme similar to He et al. (2009). The authors augment their collected dataset with the data plane measurements of CAIDA's Skitter, DIMES, and traceroutes measured from about 250 PlanetLab nodes. The resultant dataset is able to identify peerings within 223 (out of 278) IXPs which consisted of about 100% (40%) more IXPs (peerings) compared to the work of He et al. He et al. (2009).

Ager et al. (2012) rely on sFlow records from one of largest European/global IXPs as another source of information for inferring peering relationships between IXP tenants and provide insight on three fronts: (i) they outline the rich connectivity which is happening over the IXP fabric and contrast that with known private peerings which are exposed through general topology measurement studies,

(ii) present the business dynamics between participants of the IXP and providing explanation for their incentives to establish peering relationships with others, and (iii) provide the traffic matrix between peers of the IXP as a microcosm of Internet traffic. Among the set of analyses that have been conducted within the paper one could point to: (i) comparison of peering visibility from Routeviews, RIPE, LGs, and the IXPs perspective, (ii) manual label for AS types as well as the number of established peerings per member, (iii) breakdown of traffic into various protocols based on port numbers as well as the share of each traffic type among various AS types, and (iv) traffic asymmetry, ratio of used/served prefixes and geo-distance between end-points.

Khan, Kwon, Kim, and Choi (2013) utilize LG servers to provide a complementary view to Routeviews and RIPE RIR of the AS-level Internet topology. A list of 1.2k LGs (420 were operational at the time of the study) has been built by considering various sources including PeeringDB, traceroute.org, traceroute.net.ru, bgp4.as, bgp4.net, and virusnet. AS-level topologies from IRL, CAIDA's Ark, iPlane, and IRR's are used to compare the completeness of the identified AS-links. For the duration of a month **show ip bgp summary** is issued twice a week and **BGP neighbor ip advertised** is issued once a week towards all LGs which support the command. The first command outputs each neighbor's address and its associated ASN while the second command outputs the routing table of the router, consisting of reachable prefixes, next hop IP as well as the AS path towards the given prefix. AS-level connectivity graph is constructed by parsing the output of the prior commands. Using this new data source enables the authors to identify an additional 11k AS-links and about 700 new ASes.

Klöti, Ager, Kotronis, Nomikos, and Dimitropoulos (2016) perform a cross-comparison of three public IXP datasets, namely PeeringDB PeeringDB (2017), Euro-IX *European Internet Exchange Association* (2018), and PCH Packet Clearing House (2017) to study several attributes of IXPs such as location, facilities, and participants. Aside from the three aforementioned public IXP datasets, for validation purposes BGP feeds collected by PCH route collectors as well as data gathered from 40 IXP websites was used through the study. The three datasets lack common identifiers for IXPs across datasets, for this reason in a first pass IXPs are linked together through an automated process by relying on names and geo information, in the second pass linked IXPs are manually checked for correctness. The authors present one of the largest IXP information datasets at the time as a side effect of their study.

Geo coverage of each dataset is examined where the authors find relatively close coverage by each dataset except for North America region where PCH has the highest coverage. Facility location for IXPs is compared across datasets and is found that PCH lacks this information and in general facility information for IXPs is limited for other datasets. Complementarity of datasets is presented using both Jaccard and overlap index. It is found that PeeringDB and Euro-IX have the largest overlap within Europe and larger IXPs tend to have the greatest similarity across all pairs of datasets.

2.3.1.3 IXP Peerings. The studies within this section provide insight into the inner operation of IXPs and how tenants establish peerings with other ASes. Each tenant of an IXP can establish a one-to-one (bilateral) peering with other ASes of the IXP similar to how regular peerings are established. Given the large number of IXP members, a great number of peering sessions should

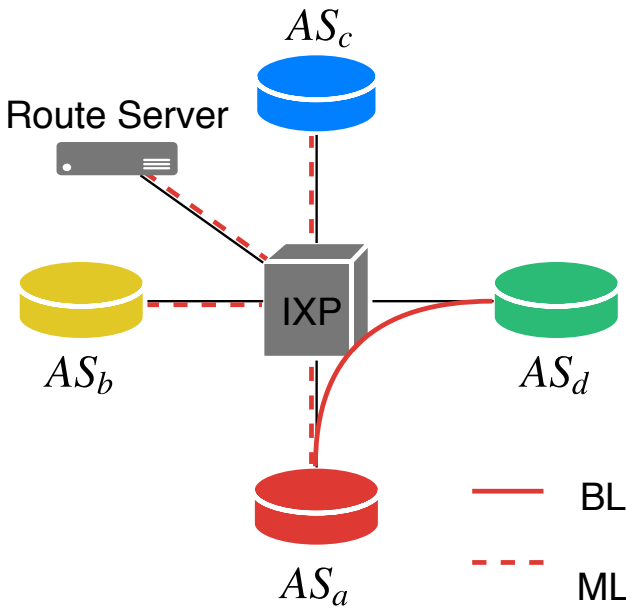


Figure 3. Illustration of an IXP switch and route server along with 4 tenant networks AS_a, AS_b, AS_c, and AS_d. AS_a establishes a bi-lateral peering with AS_d (solid red line) as well as multi-lateral peerings with AS_b and AS_c (dashed red lines) facilitated by the route server within the IXP.

be maintained over the IXP fabric. Route servers have been created to alleviate this issue where each member would establish a peering session with the route server and describe its peering preferences. This, in turn, has enabled one-to-many (multilateral) peering relationships between IXP tenants. Figure 3 illustrates an IXP with 4 tenant networks AS_a, AS_b, AS_c, and AS_d. AS_a established a bi-lateral peering with AS_d (solid red line) as well as multi-lateral peerings with AS_b and AS_c (dashed red lines) that are facilitated by the route server within the IXP. Studies within this section propose methodologies for differentiating these forms of peering relationships from each other and emphasize the importance of route servers in the operation of IXPs.

Giotsas, Zhou, Luckie, and Klaffy (2013) present a methodology to discover multilateral peerings within IXPs using the BGP communities attributes and

route server data. The BGP communities attribute which is 32bits follows specific encoding to indicate either of the following policies by each member of an IXP: (i) *ALL* routes are announced to all IXP members. (ii) *EXCLUDE* block an announcement towards a specific member, this policy is usually used in conjunction with the ALL policy. (iii) *NONE* block an announcement towards all members, and (iv) *INCLUDE* allow an announcement towards a specific member, this policy is used with the NONE policy. Using a combination of prior policies a member AS can control which IXP members receive its BGP announcements. By leveraging available LGs at IXPs and issuing router dump commands, the authors obtain the set of participating ASes and the BGP communities values for their advertised prefixes which in turn allows them to infer the connectivity among IXP participants. Furthermore, additional BGP communities values are obtained by parsing BGP feeds from Routeviews and RIPE RIS archives. Giotsas et al. infer the IXP by either parsing the first 16bits of the BGP communities attribute or by cross-checking the list of excluded ASes against IXP participants.

By combining the passive and active measurements, the authors identify 207k multilateral peering (MLP) links between 1.3k ASes. They validate their findings by finding LGs which are relevant to the identified links from PeeringDB, by testing 26k different peerings they are able to confirm 98.4% of them. Furthermore Giotsas et al. parse the peering policies of IXP members either from PeeringDB or from IXP websites which provide this information and find that 72%, 24%, and 4% of members have an open, selective, and restrictive peering policy accordingly. Participation in a route server seems to be positively correlated to a networks openness in peering. The authors present the existence of a binary pattern in terms of the number of allowed/blocked ASes where ASes either allow

or block the majority of ASes from receiving their announcements. Peering density as a representation of the percentage of established links against the number of possible links is found to be between 80%-95%.

Giotsas and Zhou (2013) expand their prior work Giotsas et al. (2013) by inferring multi-lateral peering (MLP) links between IXP tenants by merely relying on passive BGP measurements. BGP feeds are collected from both Routeviews and RIPE RIS collectors. Additionally, the list of IXP looking glasses, as well as their tenants, are gathered from PeeringDB and PCH. The authors compile a list of IXP tenants, using which the setter of each BGP announcement containing the communities attribute is determined by matching the AS path against the list of IXP tenants. If less than two ASes match against the path, no MLP link can be identified. From the two matching ASes, the AS which is closest to the prefix would be the setter, if more than two ASes match, only two ASes which have a p2p relationship according to CAIDA's AS relationship dataset are selected and the one closer to the prefix is identified as the setter. Depending on a blacklist or whitelist policy that the setter AS has chosen a list of multi-lateral peers for each setter AS is compiled.

The methodology is applied to 11 large IXP route servers; the authors find about 73% additional peering links out of which only 3% of the links are identified within CAIDA's Ark and DIMES datasets. For validation, the authors rely on IXP LGs and issue a *show ip bgp* command for each prefix. About 3k links where tested for validation and 94% of them were found to be correct.

Richter et al. (2014) outline the role and importance of route servers within IXPs. For their data, weekly snapshots of peer and master RIBs from two IXPs which exposes the multi-lateral peerings that have been happening at

the IXP are used. Furthermore, the authors have access to sFlow records which are sampled from the IXP's switching infrastructure. This dataset allows the authors to identify peerings between IXP members which have been established without the help of route servers. Using peer RIB snapshots peering relationships between IXP members as well as the symmetrical nature of it is identified. For the master RIB, Richter et al. assume peering with all members unless they find members using BGP community values to control their peering. The data plane sFlow measurements would correspond to a peering relationship if BGP traffic is exchanged between two members of the IXP. The proclivity of multi-lateral peering over bi-lateral peering is measured and found that ASes favor multi-lateral peerings with a ratio of 4:1 and 8:1 in the large and medium IXPs accordingly. Furthermore, traffic volumes transmitted over multi-lateral and bi-lateral peerings are measured and found that ASes tend to send more traffic over bi-lateral links with a ratio of 2:1 and 1:1 for the large and medium IXPs accordingly. It is found that ASes have binary behavior of either advertising all or none of their prefixes through the route server. Additionally, when ASes establish hybrid (multi and bi-lateral) peerings, they do not advertise further prefixes over their bi-lateral links. Majority of additional peerings happen over multi-lateral fabric while traffic ratios between multi(bi)-lateral peerings remain fairly consistent over the period of study.

Summary: This subsection provided an overview of researches concerned with AS-level topology. The majority of studies were concerned with the incompleteness of Internet topology graphs. These efforts lead to highlighting the importance of IXPs as central hubs of connectivity. Furthermore, various sources of information such as looking glasses, router collectors within IXPs, targeted traceroutes, RPSL entries, and traffic traces of IXPs were gleaned together to provide a more

comprehensive view of inter-AS relationships within the Internet. Lastly the importance of route servers to the inner operation of IXPs and how they enable multi-lateral peering relationships was brought into attention.

2.3.2 Router-Level Topology. Although AS-level topologies provide a preliminary view into the structure and peering relations of ASes, they merely represent virtual relationships and do not reflect details such as the number and location where these peerings are established. ASes establish interconnections with each other by placing their border routers within colos where other ASes are also present. Within these colos ASes can establish one to one peerings through private interconnections or rely on an IXPs switching fabric to establish public peerings with the IXP participants. Furthermore, some ASes extend their presence into remote colos to establish additional peerings with other ASes by relying on layer2 connectivity providers. Capturing these details can become important for accurately attributing inter-AS congestion to specific links/routers or for pin-pointing links/routers that are responsible for causing outages or disruptions within the connectivity of a physical region or network. Studies within this section aim to present methodologies to infer router-level topologies using data plane measurements in the form of traceroute. These methods would address the aforementioned shortcomings of AS-level topologies by mapping the physical entities (border routers) which are used to establish peering relations and therefore can account for multiple peering links between each AS. Furthermore, given that routers are physical entities, researchers are able to pinpoint these border routers to geo locations using various data sources and newly devised methodologies. Creating router-level topologies of the Internet can be challenging due to many reasons. First, given the span of the Internet as well as the interplay of business

relationships and routing dynamics, *traceroute* as the de-facto tool for capturing router-level topologies is only capable of recording a minute fraction of all possible paths. Routing dynamics caused by changes in each ASes route preference as well as the existence of load-balancers further complicate this task. Second, correctly inferring which set of ASes have established an inter-AS link through traceroute is not trivial due to non-standardized practices for establishing interconnections between border routers as well as several RFCs regarding the operation of routers that cause *traceroute* to depict paths that do not correspond to the forward path. Lastly, given the disassociation of the physical layer from the transport layer establishing the geolocation for the set of identified routers is not trivial. Within Section 2.2 we presented a series of platforms which try to address the first problem. The following studies summarize recent works which try to address the latter two problems.

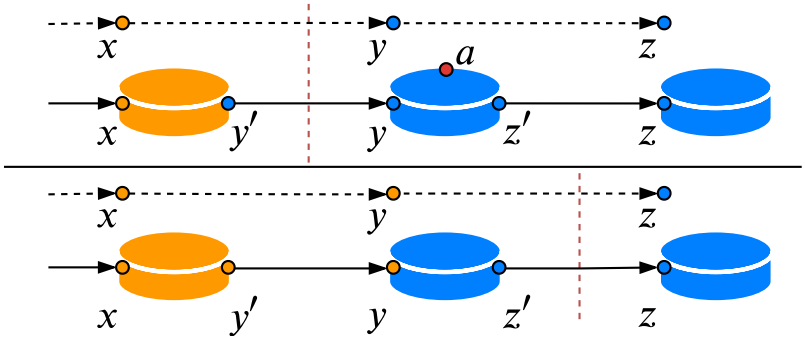


Figure 4. Illustration of address sharing for establishing an inter-AS link between border routers. Although the traceroute paths (dashed lines) are identical the inferred ownership of router interfaces and the placement of the inter-AS link differs for these two possibilities.

2.3.2.1 Peering Inference. As briefly mentioned earlier, inferring inter-AS peering relationships using traceroute paths is not trivial. To highlight this issue, consider the sample topology within Figure 4 presenting the border

routers of AS_1 and AS_2 color coded as orange and blue accordingly. This figure shows the two possibilities for address sharing on the inter-AS link. The observed traceroute path traversing these border routers is also presented at the top of each figure with dashed lines. Within the top figure AS_2 is providing the address space for the inter-AS link $(y' - y)$ while AS_1 provides the address space for the inter-AS link for the bottom figure. As we can see both of the traceroute paths are identical to each other while the ownership of router interfaces and the placement of the inter-AS link differs for these two possibilities. To further complicate the matter, a border router can respond with an interface (a in the top figure using address space owned by AS_3 color coded with red), not on the forward path of the traceroute leading to incorrect inference of an inter-AS link between AS_1 and AS_3. Lastly, the border routers of some ASes are configured to not respond to traceroute probes which restrict the chances of inferring inter-AS peerings with those ASes. The studies within this section try to address these difficulties by using a set of heuristics which are applied to a set of traceroutes that allow them to account for these difficulties.

Spring et al. (2002) done the seminal work of mapping networks of large ISPs and inferring their interconnections through traceroute probes. They make three contributions namely, (i) conducting selective traceroute probes to reduce the overall overhead of running measurements, (ii) provide an alias resolution technique to group IP address into their corresponding router, and (iii) parse DNS information to extract PoP/GEO information. Their selective probing method is composed of two main heuristics: (i) directed probing, which utilizes Routeviews data and the advertised paths to probe prefixes which are likely to cross the target network, (ii) path reduction, that avoids conducting traceroutes which would

lead to redundant paths, i.e., similar ingress or egress points. Additionally, an alias resolution technique named *Ally* is devised to group interfaces from a single network into routers. Lastly, a series of DNS parsing rules are crafted to extract geoinformation from router interface RDNS entries. The extracted geo information allows the authors to identify the PoPs of each AS. Looking glasses listed on *traceroute.org* are used to run *Rocketfuel*'s methodology to map the network of 10 ISPs including AT&T, Sprint, and Verio. The obtained maps were validated through private correspondence with network operators and by comparing the set of identified BGP neighbors with those obtainable through BGP feeds.

Nomikos and Dimitropoulos (2016) develop an augmented version of traceroute (*traIXroute*) which annotates the output path and reports whether (and at which exact hop) an IXP has been crossed along the path. The tool can operate with either traceroute or scamper as a backend. As input, *traIXroute* requires IXP membership and a list of their corresponding prefixes from PeeringDB and PCH as well as Routeviews' prefix to origin-AS mapping datasets. *traIXroute* annotates the hops of the observed path with the origin AS and tags hops which are part of an IXP prefix and also provides the mapping between an IXP address and the members ASN if such a mapping exists. Using a sliding window of size three the hops of the path are examined to find (i) hops which are part of an IXP prefix, (ii) hops which have an IXP to ASN mapping, and (iii) whether the adjacent ASes are IXP members or not. The authors account for a total of 16 possible combinations and present their assessment regarding the location of the IXP link for 8 cases that were most frequent. About 75% of observed paths matched rules which rely on IXP to ASN mapping data. The validity of this data source is looked into by using BGP dumps from routers that PCH operates within multiple IXPs. A list of IXP address

to ASN mappings was compiled by using the next hop address and first AS within the AS path from these router dumps. The authors find that 92% (93%) of the IXP to ASN mappings reported by PeeringDB (PCH) are accurate according to the BGP dumps. Finally, the prevalence of IXPs along Internet paths are measured by parsing a CAIDA Ark snapshot. About 20% of paths are reported to cross IXPs, the IXP hop on average is located on the 6th hop at the middle of the path, and only a single IXP is observed along each route which is in accordance with valley-free routing.

Luckie, Dhamdhere, Huffaker, Clark, et al. (2016) develop *bdrmap*, a method to identify inter-domain links of a target network at the granularity of individual routers by conducting targeted traceroutes. As an input to their method, they utilize originated prefixes from Routeviews and RIPE RIS, RIR delegation files, list of IXP prefixes from PeeringDB and PCH, and CAIDA's AS-to-ORG mapping dataset. Target prefixes are constructed from the BGP datasets by splitting overlapping prefixes into disjoint subnets, the first address within each prefix is targeted using *paris-traceroute*, neighbors border addresses are added to a stop list to avoid further probing within the customer's network. IP addresses are grouped together to form a router topology by performing alias resolution using Ally and Mercator. By utilizing the prefixscan tool, they try to eliminate third-party responses for cases where interfaces are responsive to alias resolution. Inferences to identify inter-AS links are done by iteratively going through a set of 8 heuristics which are designed to minimize inference errors caused by address sharing, third-party response, and networks blocking traceroute probes. Luckie et al. deploy their tool within 10 networks and receive ground truth results from 4 network operators; their method is able to identify 96-99% of inter-AS links for

these networks correctly. Furthermore, the authors compare their findings against BGP inferred relationships and find that they are able to observe between 92% - 97% of BGP links. Using a large US access network as an example, the authors study the resiliency of prefix reachability in terms of the number of exit routers and find that only 2% of prefixes exit through the same router while a great majority of prefixes had about 5-15 exit routers. Finally, the authors look at the marginal utility of using additional VPs for identifying all inter-AS links and find that results could vary depending on the target network and the geographic distribution of the VPs.

Marder and Smith (2016) devise a tool named *MAP-IT* for identifying inter-AS links by utilizing data-plane measurements in the form of traceroutes. The algorithm developed in this method requires as input the set of traceroute measurements which were conducted in addition to prefix origin-AS from BGP data as well as a list of IXP prefixes and CAIDA's AS to ORG mapping dataset. For each interface a neighbor set (N_s) composed of addresses appearing on prior (N_b) and next (N_f) hops of traceroute is created. Each interface is split into two halves, the forward and backward halves. Direct inferences are made regarding the ownership of each interface half by counting the majority ASN based on the current IP-to-AS mapping dataset. At the end of each round, if a direct inference has been made for an interface half, the other side will be updated with an indirect inference. Furthermore, within each iteration of the algorithm using the current IP-to-AS mapping, *MAP-IT* visits interface halves with direct inferences to check whether the connected AS still holds the majority, if not the inference is reduced to indirect, after visiting all interface halves any indirect inference without an associated direct inference is removed. *MAP-IT* would update the IP to AS mapping dataset

based on the current inferences and would continue this process until no further inferences are made. For verification Marder et al. use Internet2's network topology as well as a manually compiled dataset composed of DNS names for Level3 and TeleSonera interfaces. The authors investigate the effect of the hyper parameter f which controls the majority voting outcome for direct inferences and empirically find that a value of 0.5 yields the best result. Using *f=0.5 MAP-IT* has a recall of 82% - 100% and a precision of 85% - 100% for each network. The authors also look into the incremental utility of each iteration of *MAP-IT*, interestingly the majority (80%) of inferences can be made in the first round which is equivalent to making inferences based on a simple IP2AS mapping. The algorithm converges quickly after its 2nd and 3rd iterations.

Alexander et al. (2018) combine the best practices of *bdrmap* Luckie et al. (2016) and *MAP-IT* Marder and Smith (2016) into *bdrmapIT*, a tool for identifying the border routers that improves *MAP-IT*'s coverage without loosing *bdrmap*'s accuracy at identifying border routers of a single ASN. The two techniques are mainly made compatible with the introduction of "Origin AS Sets" which annotates each link between routers with the set of origin ASes from the prior hop. *bdrmapIT* relies on a two-step iterative process. During the first step, the owner of routers are inferred by counting the routers majority subsequent interfaces votes. Exceptions in terms of the casted vote for IXP interfaces, reallocated prefixes, and multi-homed routers are made to account for these cases correctly. During the second step, interfaces are annotated with an ASN using either the origin AS (if router annotation matches that of the interface) or the majority vote of prior connected routers (if router annotation differs from the interface). The iterative process is repeated until no further changes are made to the connectivity graph. The

methodology is evaluated using *bdrmap*'s ground truth dataset, as well as the ITDK dataset by removing the probes from a ground truth VP. The authors find that *bdrmapIT* improves the coverage of *MAP-IT* by up to 30% while maintaining the accuracy of *bdrmap*.

2.3.2.2 Geo Locating Routers & Remote Peering.

Historically ASes would have established their peering relations with other ASes local to their PoPs and would have relied on their upstream providers for connectivity to the remainder of the Internet. IXPs enabled ASes to establish peerings that both improved their performance due to shorter paths and reduced their overall transit costs by offload upstream traffic on p2p links instead of c2p links. With the proliferation of IXPs and their aforementioned benefits, ASes began to expand their presence not only within local IXPs but also remote ones as well. ASes would rely on layer2 connectivity providers to expand their virtual PoPs within remote physical areas. Layer3 measurements are agnostic to these dynamics and are not able to distinguish local vs. remote peering relations from each other. Researchers have tried to solve this issue by pinpointing border routers of ASes to physical locations. The association of routers to geolocations is not trivial, researchers have relied on a collection of complementary information such as geocoded embeddings within reverse DNS names or by constraining the set of possible locations through colo listings offered by PeeringDB and similar datasets. In the following, we present a series of recent studies which tackle this unique issue.

Castro et al. (2014) present a methodology for identifying remote peerings, where two networks interconnect with each other via a layer-2 connectivity provider. Furthermore, they derive analytical conditions for the economic viability of remote peering versus relying on transit providers. Levering PeeringDB, PCH,

and information available on IXP websites a list of IXP's as well as their tenants, prefixes and interface to member mapping is obtained. For this study, IXPs which have at least one LG or RIPE NCC probe (amounting to a total of 22) are selected. By issuing temporally spaced probes towards all of the identified interfaces within IXP prefixes and filtering interfaces which either do not respond frequently or do not match an expected maximum TTL value of 255 or 64 a minimum RTT value for each interface is obtained. By examining the distribution of minimum RTT for each interface, a conservative threshold of 10ms is selected to consider an interface as remote. A total of 4.5k interfaces corresponding to 1.9k ASes in 22 IXPs are probed in the study. The authors find that 91% of IXPs have remote peering while 285 ASes have a remote interface. Findings including RTT measures as well as remote labels for IXP members were confirmed for TorIX by the staff. One month of Netflow data captured at the border routers of RedIRIS (Spain's research and education network) is used to examine the amount of inbound and outbound traffic between RedIRIS and its transit providers, using which an upper bound for traffic which can be offloaded is estimated. Furthermore, the authors create a list of potential peers (2.2k) which are reachable through Euro-IX, these potential peers are also categorized into different groups based on their peering policy which is listed on PeeringDB. Considering all of the 2.2k networks RedIRIS can offload 27% (33%) of its inbound (outbound) traffic by remotely peering with these ASes. Through their analytical modeling, the authors find that remote peering is viable for networks with global traffic as well as networks which have higher ratios of traffic-independent cost for direct peering compared to remote peering such as networks within Africa.

Giotsas, Smaragdakis, Huffaker, Luckie, and claffy (2015) attempt to obtain a peering interconnection map at the granularity of colo facilities. Authors gather AS to facility mapping information from PeeringDB as well as manually parsing this information for a subset of networks from their websites. IXP lists and members were compiled by combining data from PeeringDB, PCH, and IXP websites. For data-plane measurements, the authors utilize traceroute data from RIPE Atlas, iPlane, CAIDA's Ark, and a series of targeted traceroutes conducted from looking glasses. The authors annotate traceroute hops with their corresponding ASN and consider the segment which has a change in ASN as the inter-AS link. Using the colo-facility listing obtained in the prior step the authors produce a list of candid facilities for each inter-AS link which can result in three cases: (i) a single facility is found, (ii) multiple facilities match the criteria, or (iii) no candid facility is found. For the latter two cases, the author's further constraint the search space by either benefiting from alias resolution results (two alias interfaces should reside in the same facility) or by conducting further targeted probes which are aimed at ASNs that have a common facility with the owner AS of the interface in question. The methodology is applied to five content providers (Google, Yahoo, Akamai, Limelight, and Cloudflare) and five transit networks (NTT, Cogent, DT, Level3, and Telia). The authors present the effect of each round of their constrained facility search (CFS) algorithm's iteration (max iteration count of 100), the majority of pinned interfaces are identified up to the 40th iteration with RIPE probes providing a better opportunity for resolving new interfaces. The authors find that DNS-based pinning methods are able to identify only 32% of their findings. The authors also cross-validate their findings using direct feedback from network admins, BGP communities attribute, DNS records,

and IXP websites with 90% of the interfaces being pinned correctly and for the remainder, the pinning accuracy was correct at a metro granularity.

Nomikos et al. (2018) present a methodology for identifying remote peers within IXPs, furthermore they apply their methodology to 30 large IXPs and characterize different aspects of the remote peering ecosystem. They define an IXP member as a remote peer if it is not physically connected to the IXPs fabric or reaches the IXP through a reseller. The development of the methodology and the heuristics used by the authors are motivated by a validation dataset which they obtain through directly contacting several IXP operators. A collection of 5 heuristics are used in order to infer whether an IXP member is peering locally or remotely these heuristics in order of importance are: (i) the port capacity of a customer, (ii) latency measurements from VPs within IXPs towards customer interfaces, (iii) colocation locations within an RTT radius, (iv) multi-IXP router inferences by parsing traceroutes from publicly available datasets and corroborating the location of these IXPs and whether the AS in question is local to any of them, and (v) identifying private peerings (by parsing public traceroute measurements) between the target AS and one or more local IXP members is used as a last resort to infer whether a network is local or remote to a given IXP. The methodology is applied to 30 large IXPs, and the authors find that a combination of RTT and colo listings to be the most effective heuristics in inferring remote peers. Overall 28% of interfaces are inferred to be peering remotely and for 90% of IXPs. The size of local and remote ASes in terms of customer cone is observed to be similar while hybrid ASes tend to have larger network sizes. The growth of remote peering is investigated over a 14 month period, and the authors find that the number of remote peers grew twice as fast as the number of local peers.

Motamedi et al. (2019) propose a methodology for inferring and geolocating interconnections at a colo level. The authors obtain a list of colo facility members from PeeringDB and colo provider webpages. A series of traceroutes towards the address space of prior steps ASes are conducted using available measurement platforms such as looking glasses and RIPE Atlas nodes in the geo proximity of the targeted colo. *tracerotue* paths are translated to a router-level connectivity graph using alias resolution and a set of heuristics based on topology constraints. The authors argue that a router-level topology coupled with the prevalence of observations allows them to account for *traceroute* anomalies and they are able to infer the correct ASes involved in each peering. To geolocate routers, an initial set of *anchor* interfaces with a known location is created by parsing reverse DNS entries for the observed router interfaces. This information is propagated/expanded through the router-level graph by a Belief Propagation algorithm that uses a set of co-presence rules based-on membership in the same alias set and latency difference between neighboring interfaces.

Summary: while traceroutes have been historically utilized as a source of information to infer inter-AS links, the methodologies did not correctly account for the complexities of inferring BGP peerings from layer-3 probes. The common practice of simply mapping interface addresses along the path to their origin-AS based on BGP data does not account for the visibility of BGP collectors, address sharing for establishing inter-AS links, third-party responses of TTL expired messages by routers, and unresponsive routers or firewalled networks along the traceroute path. The presented methodologies within this section attempt to account for these difficulties by corroborating domain knowledge for common networking practices and relying on a collection of traceroute paths and their corresponding

router view (obtained by using alias resolution techniques) to make accurate inferences of the entities which are establishing inter-AS links. Furthermore, pinpointing routers to physical locations was the key enabler for highlighting remote peerings that are simply not visible from an AS-level topology.

2.3.3 PoP-Level Topology. PoP-level topologies present a middle ground between AS-level and router-level topologies. A PoP-level graph presents the points of presence for one or many networks. These topologies inherently have geo information at the granularity of metro areas embedded within. They have been historically at the center of focus as many ASes disclose their topologies at a PoP level granularity and do not require detailed information regarding each individual router and merely represent a bundle of routers within each PoP as a single node. They have lost their traction to router-level topologies that are able to capture the dynamics of these topologies in addition to providing finer details of information. Regardless of this, due to the importance of some ASes and their centrality in the operation of today's Internet, several studies Schlinker et al. (2017); Wohlfart, Chatzis, Dabanoglu, Carle, and Willinger (2018); Yap et al. (2017) outlining the internal operation of these ASes within each PoP have emerged. These studies offer insight into the challenges these ASes face for peering and serving the vast majority of the Internet as well as the solutions that they have devised.

Cunha et al. (2016) develop *Sibyl*, a system which provides an expressive interface that allows the user to specify the requirements for the path of a traceroute, given the set of requirements *Sibyl* would utilize all available vantage points and rely on historical data to conduct a traceroute from a given vantage point towards a specific destination that is most likely to satisfy the users

constraints. Furthermore, given that each vantage point has limited probing resources and that concurrent requests can be made, *Sibyl* would pick source-destination pairs which optimize for resource utilization. *Sibyl* combines PlanetLab, RIPE Atlas, traceroute servers accessible through looking glasses, DIMES, and Dasu measurement platforms to maximize its coverage. Symbolic regular expressions are used for the query interface where the user can express path properties such as the set of traversed ASes, cities, and PoPs. The likelihood of each source-destination pair matching the required path properties is calculated using a supervised machine learning technique (RuleFit) which is trained based on prior measurements and is continuously updated based on new measurements. Resource utilization optimization is addressed by using a greedy algorithm, *Sibyl* chooses to issue traceroutes that fit the required budget and that have the largest marginal expected utility based on the output of the trained model.

Schlinker et al. (2017) outline Facebook's edge fabric within their PoPs by utilizing an SDN based system that alters BGP local-pref attributes to utilize alternative paths towards specific prefixes better. The work is motivated by BGP's shortcomings namely, lack of awareness of link capacities and incapability to optimize path selections based on various performance metrics. More specifically BGP makes its forwarding decisions using a combination of AS-path length and the local-perf metric. Facebook establishes BGP connections with other ASes through various means namely, private interconnections, public peerings through IXPs, and peerings through router servers within IXPs. The authors report that the majority of their interconnections are established through public peerings while the bulk of traffic is transmitted over the private links. The later reflects Facebook's preference to select private peerings over public peerings while peerings established through

route servers have the lowest priority. Furthermore, the authors observe that for all PoPs except one, all prefixes have at least two routes towards each destination prefix. The proposed solution isolates the traffic engineering per PoP to simplify the design, the centralized SDN controller within each PoP gathers router RIB tables through a BMP collector. Furthermore, traffic statistics are gathered through sampled sFlow or IPFIX records. Finally, interface information is periodically pulled by SNMP. The collector emulates BGP's best path selection and projects interface utilization. For overloaded interfaces prefixes with alternative routes are selected, an alternative route is selected based on a set of preferences. The output of this step generates a set of route overrides which are enforced by setting a high local-pref value for them. The authors report that their deployed system detours traffic from 18% of interfaces. The median of detour time is 22 minutes and about 10% of detours last as long as 6 hours. The detoured routes resulted in 45% of the prefixes achieving a median latency improvement of 20ms while 2% of prefixes improved their latency by 100ms.

Yap et al. (2017) discuss the details of Espresso, an application-aware routing system for Google's peering edge routing infrastructure. Similar to the work of Schlinker et at. Schlinker et al. (2017) Espresso is motivated by the need for a more efficient (both technically and economically) edge peering fabric that can account for traffic engineering constraints. Unlike the work of Schlinker et al. Schlinker et al. (2017) Espresso maintains two layers of control plane one which is localized to each PoP while the other is a global centralized controller that allows Google to perform further traffic optimizations. Espresso relies on commodity MPLS switches for peering purposes, traffic between the switches and servers are encapsulated in IP-GRE and MPLS headers. IP-GRE header encodes the correct

switch, and the MPLS header determines the peering port. The global controller (GC) maintains an egress map that associates each client prefix and PoP tuple to an edge router/switch and egress port. User traffic characteristics such as throughput, RTT, and re-transmits are reported at a /24 granularity to the global controller. Link utilization, drops, and port speeds are also reported back to the global controller. A greedy algorithm is used by the GC to assign traffic to a candid router port combination. The greedy algorithm starts by making its decisions using traffic priority metrics and orders its available options based on BGP policies, user traffic metrics, and the cost of serving on a specific link. Espresso has been incrementally deployed within Google and at the time of the study was responsible for serving about 22% of traffic. Espresso is able to maintain higher link utilization while maintaining low packet drop rates even for fully utilized links (95% less than 2.5%). The authors report that the congestion reaction feature of the GC results in higher goodput and mean time between re-buffers for video traffic.

Wohlfart et al. (2018) present an in-depth study of the connectivity fabric of Akamai at its edge towards its peers. The authors account 3.3k end-user facing (EUF) server deployments with varying size and capabilities which are categorized into four main groups. Two of these groups have Akamai border routers and therefore establish explicit peerings with peers and deliver content directly to them while the other two groups are hosted within another ASes network and are responsible for delivering content implicitly to other peers. Customers are redirected to the correct EUF server through DNS, the mapping is established by considering various inputs including BGP feeds collected by Akamai routers, user performance metrics, and link cost information. To analyze Akamai's peering fabric, the authors rely on proprietary BGP snapshots obtained from Akamai

routers and consist of 3.65M AS paths and about 1.85M IPv4 and IPv6 prefixes within 61k ASes (ViewA). As a point of comparison, a combination of daily BGP feeds from Routeviews, RIPE RIS, and PCH consisting of 21.1M AS paths and 900k prefixes within 59k ASes is used (ViewP). While at an AS level both datasets seem to have a relatively similar view, ViewA (ViewP) observes 1M (0.1M) prefixes the majority of which are prefixes longer than /25. Only 15% of AS paths within ViewP are observed by ViewA which suggests that a large number of AS paths within ViewP are irrelevant for the operation of Akamai. Wohlfart et al. report 6.1k unique explicit peerings between Akamai and its neighbors by counting the unique number of next-hop ASN from the Akamai BGP router dumps. About 6k of these peerings happen through IXPs while the remainder are established through PNIs. In comparison, only 450 peerings between Akamai and other ASes are observed through ViewP. Using AS paths within ViewP the authors report about 28k implicit peers which are within one AS hop from Akamai's network. Lastly, the performance of users sessions are looked into by utilizing EUF server logs containing the clients IP address, throughput, and a smoothed RTT value. The performance statistics are presented for two case studies (i) serving a single ISP and (ii) serving customers within 6 distinct metros. Overall 90% of traffic is coming from about 1% of paths and PNIs are responsible for delivering the bulk of traffic and PNIs and cache servers within eyeball ASes achieve the best performance regarding RTT.

Nur and Tozal (2018) study the Internet AS-level topology using a multigraph representation where AS pairs can have multiple edges between each other. Traceroute measurements from CAIDA's Ark and iPlane projects are collected for this study. For IP to AS mapping Routeviews' BGP feed is utilized.

Next hop addresses for BGP announcements are extracted from Routeviews as well as RIPE RIS. For mapping IP addresses to their corresponding geo-location various data sources have been employed namely, (i) UNDNS for DNS parsing, (ii) DB-IP, (iii) Maxmind GeoLite2 City, and (iv) IP2Location DB5 Lite.

Each ASes border interface is identified by tracking ASN changes along the hops of each traceroute. Each cross border interface X-BI is geolocated to the city in which it resides by applying one of the following methods in order of precedence: (i) relying on UNDNS for extracting geoinformation from reverse DNS names, (ii) majority vote along three (DB-IP, Maxmind, and IP2Location) IP to GEO location datasets, (iii) sandwich method where an unresolved IP between two IPs in the same geolocation is mapped to the same location, (iv) RTT based geo locating which relies on the geolocation of prior or next hops of an unresolved address that have a RTT difference smaller than 3 ms for mapping them to the same location, and (v) if all of the prior methods fail Maxmind's output is used for mapping the geolocation of the X-BI. The set of inter-AS links resulting from parsing traceroutes is augmented by benefiting from BGP data. If an AS relationship exists between two ASes but is missing from the current AS-level graph and all identified X-BIs corresponding to these ASes are geolocated to a single city, a link will be added to the AS-level topology graph under the assumption that this is the only possible location for establishing an interconnection between these two ASes.

The inferred PoP nodes in the AS graph are validated for major research networks as well as several commercial ISPs. The overlap of identified PoPs is measured for networks which have publicly available PoP-level maps. The maps align with the set of identified cities by X-AS with deviations in terms of number of PoPs per city. This is a limitation of X-AS as it is only able to identify one

PoP per city. Identified AS-links are compared against CAIDA's AS relationships dataset, the percentage of discrepancy for AS links of each AS is measured. For 78% of ASes, the maps agree with each other completely, and the average link agreement is about 85% for all ASes. Various properties of the resulting graph are analyzed in the paper, the authors find that the number of X-BI nodes per AS, X-BI nodes degree, and AS degree all follow a power law distribution.

Summary: PoP-level topologies can offer a middle ground between router-level and AS-level topologies offering an understanding of inter-AS peering relationships while also being able to distinguish instances of these peerings happening at various geo-locations/PoPs. Additionally, we reviewed studies that elaborate on the faced challenges as well as the devised solutions for content provider (Google, Facebook) and CDN (Akamai) networks which are central to the operation of today's Internet.

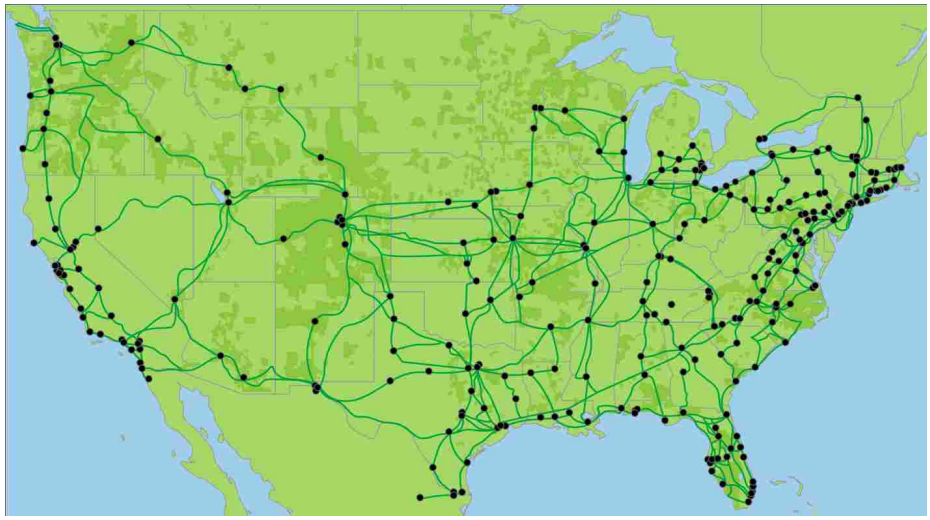


Figure 5. Fiber optic backbone map for CenturyLink's network in continental US. Each node represents a PoP for CenturyLink while links between these PoPs are representative of the fiber optic conduits connecting these PoPs together. Image courtesy of CenturyLink.

2.3.4 Physical-Level Topology.

This subsection is motivated by the works of Knight, Nguyen, Falkner, Bowden, and Roughan (2011) and Durairajan, Ghosh, Tang, Barford, and Eriksson (2013); Durairajan, Sommers, and Barford (2014); Durairajan, Sommers, Willinger, and Barford (2015) which presented the groundwork for having a comprehensive physical map of the Internet consisting of edges corresponding to fiber optic cables providing connectivity between metro areas and PoPs as nodes within these topologies. A sample of this topology for CenturyLink's fiber-optic backbone network within the continental US is presented in Figure 5. Physical maps were mostly neglected by the Internet topology community mainly due to two reasons: (i) the scarcity of well-formatted information and (ii) the complete disassociation of physical layers from probes conducted within higher layers of the TCP/IP stack. The following set of papers try to address the former issue by gathering various sources of information and compiling them into a unified format.

Knight et al. (2011) present the Internet topology Zoo which is a collection of physical maps of various networks within the Internet. The authors rely on ground truth data publicly provided by the network operators on their websites. These maps are presented in various formats such as static images or flash objects. The authors transcribe all maps using yEd (a graph editor and diagraming program) into a unified graph specification format (GML) and annotate nodes and links with any additional information such as link speed, link type, longitude, and latitudes that is provided by these maps. Each map and its corresponding network is classified as a backbone, testbed, customer, transit, access or internet exchange based on the properties of their network. For example, backbone networks should connect at least two cities together while access networks should provide

edge access to individuals. A total of 232 networks are transcribed by the authors. About 50% of networks are found to have more than 21 PoPs and each of these PoPs have an average degree of about 3. Lastly similar to Gregori et al. (2011) the core density of networks is examined by measuring the 2-core size of networks. A wide degree of 2-core sizes ranging from 0 (tree-like networks) to 1 (densely connected core with hanging edges) are found within the dataset.

Durairajan et al. (2013) create a map of the physical Internet consisting of nodes representing colocation facilities and data-centers, links representing conduits between these nodes and additional metadata related to these entities. The authors rely on publicly available network maps (images, Flash objects, Google Maps overlays) provided by ASes. The methodology for transcribing images consists of 5 steps: (i) capturing high-resolution sub-images, (ii) patching sub-images into a composite image, (iii) extracting a link image using color masking techniques, (iv) importing link image into ArcGIS using geographic reference points, and (v) using link vectorization in ArcGIS to convert links into vectors. Given that each map has a different geo resolution, different scores are attributed to nodes with lat/lon or street level, city, and state having a corresponding score of 1.0, 0.75, 0.5. All maps have at least city level resolution with about 20% of nodes having lat/lon or street level accuracy.

Durairajan et al. (2014) work is motivated by two research questions: (i) how do physical layer and network layer maps compare with each other? and (ii) how can probing techniques be improved to reveal a larger portion of physical infrastructure? For physical topologies, the authors rely on maps which are available from the Internet Atlas project. From this repository the maps for 7 Tier-1 networks and 71 non-Tier-1 networks which are present in North America are

gathered, these ASes collectively consist of 2.6k PoPs and 3.6k links. For network layer topologies, traceroutes from the CAIDA Ark project during the September 2011 to Match 2013 period are used. Additionally DNS names for router interfaces are gathered from the IPv4 Routed /24 DNS Names Dataset which includes the domain names for IP addresses observed in the CAIDA Ark traceroutes. Traceroute hops are annotated with their corresponding geo information (extracted with DDeC) as well as the AS number which is collected from TeamCymru's service. Effects of vantage point selection on node identification are studied by employing public traceroute servers. Different modalities depending on the AS ownership of the traceroute server and the target address are considered ([VP_in, t_in], [VP_in, t_out], [VP_out, t_in]). Their methodology (POPsicle) chooses VPs based on geo proximity towards the selected targets and along the pool of destinations, those which have a square VP to destination distance greater than the sum of squares of the distance between target VP and destination are selected to create a measurement cone. For this study 50 networks that have a comprehensive set of geo-information for their physical map are considered. Out of these 50 networks, 21 of them do not have any geo information embedded in their DNS names. Furthermore, 16 ASes were not observed in the Ark traces. This results in 13 ASes out of the original 50 which have both traces and geo-information in the network layer map. POPsicle was deployed in an IXP (Equinix Chicago) to identify the PoPs of 10 tenants. Except for two networks, POPsicle was able to identify all known PoPs of these networks. Furthermore, POPsicle was evaluated by targeting 13 ISPs through Atlas probes which were deployed in IXPs, for all of these ISPs POPsicle was able to match or outperform Ark and Rocketfuel. Furthermore for 8 of these ISPs POPsicle found all or the majority of PoPs present in Atlas maps.

Durairajan et al. (2015) obtain the long-haul fiber network within the US and study its characteristics and limitations. For the construction of the long-haul fiber map, Durairajan et al. rely on the Internet Atlas project Durairajan et al. (2013) as a starting point and confirm the geo-location or sharing of conduits through legal documents which outline laying/utilization of infrastructure. The methodology consists of four steps: (i) using Internet Atlas maps for tier-1 ASes that have geo-coded information, a basic map is constructed, (ii) the geolocation of nodes and links for the map is confirmed through any form of legal document which can be obtained, (iii) the map is augmented with additional maps from large transit ASes which lack any geo-coded information, (iv) the augmented map is once again confirmed through any legal document that would either confirm the geolocation of a node/link or would indicate conduit sharing with links that have geo-coded information. The long-haul fiber map seems to be physically aligned with roadway and railway routes, the authors use the polygon overlap feature of ArcGIS to compare the overlap of these maps and find that most often long-hauls run along roadways. The authors also assess shared conduit risks, for this purpose they construct a conduit sharing matrix were rows are ASes and columns are conduits the value within each row indicates the number of ASes which are utilizing that conduit. Out of 542 identified conduits about 90% of them are shared by at least one other AS. Using the risk matrix the hamming distance for each AS pair is measured to identify ASes which have similar risk profiles. Using traceroute data from Edgescape and parsing geoinformation in domain names the authors infer which conduits were utilized by each traceroute and utilize the frequency of traceroutes as a proxy measure of traffic volume. Finally a series of risk mitigation analysis are conducted namely: (i) the possibility of increasing network robustness

by utilizing available conduits or by peering with other networks is investigated for each AS (ii) increasing network robustness through the addition of additional k links is measured for each network, and lastly (iii) possibility for improving latency is investigated by comparing avg latencies against right of way (ROW), line of sight (LOS), and best path delays.

Summary: the papers within this sub-section provided an overview of groundbreaking works that reveal physical-level topologies of the Internet. The researchers gathered various publicly available maps of ASes as well as legal documents pertaining to the physical location of these networks to create a unified, well-formatted repository for all these maps. Furthermore, the applicability of these maps towards the improvement of targeted probing methodologies and the possibility of improving and provisioning the infrastructure of each network is investigated. Although the interplay of routing on top of these physical topologies is unknown and remains as an open problem, these physical topologies provide complementary insight into the operation of the Internet and allow researchers to provision or design physical infrastructure supporting lower latency Internet access or to measure the resiliency of networks towards natural disasters.

2.4 Implications & Applications of Network Topology

This section will provide an overview of the studies which rely on Internet topology to provide additional insight regarding the performance, resiliency, and various characteristics of the Internet. The studies which are outlined in this section look into various properties of the Internet including but not limited to: path length both in terms of router and AS hops, latency, throughput, packet loss, redundancy, and content proximity. In a more broad sense, we can categories these studies into three main groups: *(i) studying performance characteristics of the*

Internet, *(ii) studying resiliency of the Internet*, and *(iii) classifying the type of inter-AS relationships between ASes*. Depending on the objective of the study one or more of the aforementioned properties of the Internet could be the subject which these studies focus on. Each of these studies would require different resolutions of Internet topology. As outlined in Section 2.3 obtaining a one to one mapping between different resolutions is not always possible. For example, each AS link can correspond to multiple router level links while each router level link can correspond to multiple physical links. For this reason, each study would rely on a topology map which better captures the problems objectives. As an example, studying the resiliency of a transit ASes backbone to natural disasters should rely on a physical map while performing the same analyses using an AS-level topology could lead to erroneous conclusions given the disassociation of ASes to physical locations. While on the other hand studying the reachability and visibility of an AS through the Internet would require an AS-level topology and conducting the same study using a fiber map would be inappropriate as the interplay of the global routing system on top of this physical map is not known. The remainder of this section would be organized into three sub-sections presenting the set of studies which focus on the *(i) Internet performance*, *(ii) Internet resiliency*, and *(iii) AS relationship classification*. Furthermore, each sub-section would further divide the studies based on the granularity of the topology which is employed.

2.4.1 Performance. Raw performance metrics such as latency and throughput can be conducted using end-to-end measurements without any attention to the underlying topology. While these measurements can be insightful on their own, gaining a further understanding of the root cause of subpar performance often requires knowledge of the underlying topology. For example,

high latency values reported through end-to-end measurements can be a side effect of many factors including but not limited to congestion, a non-optimal route, an overloaded server, and application level latencies. Many of these underlying causes can only be identified by a correct understanding of the underlying topology. Congestion can happen on various links along the forward and reverse path, identifying the faulty congested link or more specifically the inter-AS link requires a correct mapping for the traversed topology. Expanding infrastructure to address congestion or subpar latency detected through end-to-end measurements is possible through an understanding of the correct topology as well as the interplay of routing on top of this topology. In the following Section, we will present studies that have relied on router, AS, and physical level topologies to provide insight into various network performance related issues.

2.4.1.1 AS-Level Topology. Studies in this section rely on BGP feeds as well as traceroute probes that have been translated to AS paths to study performance characteristics such as increased latency and path lengths due to insufficient network infrastructure within Africa Fanou, Francois, and Aben (2015); Gupta et al. (2014), path stability and the latency penalties due to AS path changes Green, Lambert, Pelsser, and Rossi (2018), IXPs centrality in Internet connectivity as a means for reducing path distances towards popular content Chatzis, Smaragdakis, Böttger, Krenc, and Feldmann (2013), and estimating traffic load on inter-AS links through the popularity of traversed paths Sanchez et al. (2014).

Chatzis et al. (2013) demonstrate the centrality of a large European IXP in the Internet's traffic by relying on sampled sFlow traces captured by the IXP operator. Peering relationships are identified by observing BGP as well as regular

traffic being exchanged between tenant members. The authors limit their focus to web traffic as it constitutes the bulk of traffic which is observed over the IXP's fabric. Endhost IP addresses are mapped to the country which they reside in by using Maxmind's IP to GEO dataset. The authors observe traffic from nearly every country (242 out of 250). While tenant ASes generated the bulk of traffic, about 33% of traffic originated from ASes which were one or more hops away from the IXP. The authors find that recurrent IP addresses generate about 60% of server traffic. Finally, the authors highlight the heterogeneity of AS traffic by identifying servers from other ASes which are hosted within another AS. Heterogeneous servers are identified by applying a clustering algorithm on top of the SOA records of all observed IP addresses. Lastly, the share of heterogeneous traffic on inter-AS links is presented for Akamai and Cloudflare. It is found that about 11% (54%) of traffic (servers) are originated (located) within 3rd-party networks.

Sanchez et al. (2014) attempt to characterize and measure inter-domain traffic by utilizing traceroutes as a proxy measure. Traceroute probes towards random IP addresses from the Ono BitTorrent extension are gathered over two separate months. Ground truth data regarding traffic volume is obtained from two sources: (i) sampled sFlows from a large European IXP and (ii) link utilization for the customers of a large ISP presenting the 95th percentile of utilization using SNMP.

AS-link traversing paths (ALTP) are constructed by mapping each hop of traceroutes to their corresponding ASN. For each ALTP-set a relative measure of link frequency is defined which represents the cardinality of the link to the sum of cardinalities of all links in that set. This measure is used as a proxy for traffic volume. The authors measure different network syntax metrics namely:

connectivity, control value, global choice, and integration for the ALTP-sets which have common links with their ground truth traffic data. r^2 is measured for regression analysis of the correlation between network syntax metrics and traffic volume. ALTP-frequency shows the strongest correlation with r^2 values between 0.71 - 0.97 while the remainder of metrics also show strong and very-strong correlations. The authors utilize the regression model to predict traffic volume using ALTP-frequency as a proxy measure. Furthermore Sanchez et al. demonstrate that the same inferences cannot be made from a simple AS-level connectivity graph which is derived from BGP streams. Finally, the authors apply the same methodology to CAIDA's Ark dataset and find similar results regarding the correlation of network syntax metrics and traffic volume.

Gupta et al. (2014) study circuitous routes in Africa and their degrading effect on latency. Circuitous routes are between two endpoints within Africa that traverse a path outside of Africa, i.e. the traversed route should have ideally remained within Africa but due to sub-par connectivity has detoured to a country outside of Africa. Two major datasets are used for the study, (i) BGP routing tables from Routeviews, PCH, and Hurricane Electric, and (ii) periodic (every 30 minutes) traceroute measurements from BISmark home routers towards MLab servers, IXP participants, and Google cache servers deployed across Africa. Traceroute hops are annotated with their AS owner and inter-AS links are identified with the observation of ASN changes along the path. Circuitous routes are identified by relying on high latency values for the given path. Latency penalty is measured as the ratio of path latency to the best case latency between the source node and a node in the same destination city. The authors find two main reasons for paths with high latency penalty values namely, (i) ASes along the path are not

physically present at a local IXP, or (ii) the ASes are present at a geographically closeby IXP but do not peer with each other due to business preferences.

Fanou et al. (2015) study Internet topology and its characteristics within Africa. By expanding RIPE's Atlas infrastructure within African countries, the authors leverage this platform to conduct traceroute campaigns with the intention of uncovering as many as possible AS paths. To this end, periodic traceroutes were ran between all Atlas nodes within Africa. These probes would target both IPv4 and IPv6 addresses if available. Traceroute hops were mapped to their corresponding country by leveraging six public datasets, namely OpenIPMap, MaxMind, Team Cymru, AFRINIC DB, Whois, and reverse DNS lookup. Upon disagreement between datasets, RIPE probes within the returned countries were employed to measure latency towards the IP addresses in question, the country with the lowest latency was selected as the host country. Interface addresses are mapped to their corresponding ASN by utilizing Team Cymru's IP to AS service TeamCymru (2008), using the augmented traceroute path the AS path between the source and destination is inferred. Using temporal data the preference of AS pairs to utilize the same path is studied, 73% (82%) of IPv4 (IPv6) paths utilize a path with a frequency higher than 90%. Path length for AS pairs within west and south Africa are studied, with southern countries having a slightly shorter average path of 4 compare to 5. AS path for pairs of addresses which reside within the same country in each region is also measured where it's found that southern countries have a much shorter path compared to pairs of addresses which are in the same western Africa countries (average of 3 compared to 5). AS-centrality (percentage of paths which AS appears in and is not the source or destination) is measured to study transit roles of ASes. Impact of intercontinental transit on end-to-end

delay is measured by identifying the IP path which has the minimum RTT. It is found that intercontinental paths typically exhibit higher RTT values while a small fraction of these routes still have relatively low RTT values ($< 100ms$) and are attributed to inaccuracies in IP to geolocation mapping datasets.

Green et al. (2018) leverage inter-AS path stability as a measure for conducting Internet tomography and anomaly detection. Path stability is analyzed by the stability of a *primary path*. The primary path of router r towards prefix p is defined as the most prevalent preferred path by r during the window time-frame of W. Relying on 3 months of BGP feeds from RIPE RIS' LINX collector it is demonstrated that 85% (90%) of IPv4 (IPv6) primary paths are in use for at least half of the time. Any deviation from the primary path are defined as pseudo-events which are further categorized into two groups: (i) transient events where a router explores additional paths before reconverging to the primary path, and (ii) structural events where a router consistently switches to a new primary path. For each pseudo-event, the duration and set of new paths that were explored are recorded. About 13% of transient pseudo-events are found to be longer than an hour while 12% of structural pseudo-events last less than 7 days. The number of explored paths and the recurrence of each path is measured for pseudo-events. It is found that MRAI timers and route flap damping are efficient at regulating BGP dynamics. However, these transient events could be recurrent and require more complex mechanisms in order to be accounted for. For anomaly detection about 2.3k AS-level outages and hijack events reported by BGPmon during the same period of the study are used as ground truth. About 84% of outages are detected as pseudo-events in the same time window while about 14% of events the detection time was about one hour earlier than what BGPmon reported. For hijacks, the

announced prefix is looked-up amongst pseudo-events if no match is found less specific prefixes are used as a point of comparison with BGPmon. For about 82% of hijacking events, a matching pseudo-event was found, and the remainder of events are tagged as explicit disagreements.

2.4.1.2 Router-Level Topology.

With the rise of peering disputes highlighted by claims of throttling for Netflix's traffic access to unbiased measurements reflecting the underlying cause of subpar performance seems necessary more than before. Doing so would require a topology map which captures inter-AS links. The granularity of these links should be at the router level since two ASes could establish many interconnections with each other, each of which could exhibit different characteristics in terms of congestion. As outlined in Section 2.3 various methodologies have been presented that enable researchers to infer the placement of inter-AS links from data plane measurements in the form of traceroutes. A correct assessment of the placement of inter-AS links is necessary to avoid attributing intra-AS congestion to inter-AS congestion, furthermore incorrectly identifying the ASes which are part of the inter-AS link could lead to attributing congestion to incorrect entities.

Dhamdhere et al. (2018) rely on prior techniques Luckie et al. (2016) to infer both ends of an interconnect link and by conducting time series latency probes (TSLP) try to detect windows of time where the latency time series deviates from its usual profile. Observing asymmetric congestion for both ends of a link is attributed to inter-AS congestion. The authors deploy 86 vantage points within 47 ASes on a global scale. By conducting similar TSLP measurements towards the set of identified inter-AS links over the span of 21 months starting at March 2016, the authors study congestion patterns between various networks and their

upstream transit providers as well the interconnections they establish with content providers. Additionally, the authors conduct throughput measurements using the Network Diagnostic Tool (NDT) M-Lab (2018) as well as SamKnows SamKnows (2018) throughput measurements of Youtube servers and investigate the correlation of inter-AS congestion and throughput.

Chandrasekaran, Smaragdakis, Berger, Luckie, and Ng (2015) utilize a large content delivery networks infrastructure to assess the performance of the Internet's core. The authors rely on about 600 servers spanning 70 countries and conduct pairwise path measurements in both forward and reverse directions between the servers. Furthermore, AS paths are measured by translating router hop interfaces to their corresponding AS owner, additionally inter-AS segments are inferred by relying on a series of heuristics developed by the authors based on domain knowledge and common networking practices. Latency characteristics of the observed paths are measured by conducting periodic ping probes between the server pairs. Consistency and prevalence of AS paths for each server pair are measured for a 16 month period. It is found that about 80% of paths are dominant for at least half of the measurement period. Furthermore, about 80% of paths experience 20 or fewer route changes during the 16 month measurement period. The authors measure RTT inflation in comparison to optimal AS paths and find that sub-optimal paths are often short-lived although a small number (10%) of paths experience RTT inflation for about 30% of the measurement period. Effects of congestion on RTT inflation are measured by initially selecting the set of server pairs which experience RTT inflation using ping probe measurements while the first segment that experiences congestion is pinpointed by relying on traceroute measurements which are temporally aligned with the ping measurements. The

authors report that most inter and intra-AS links experience about 20 to 30 ms of added RTT due to congestion.

Chiu, Schlinker, Radhakrishnan, Katz-Bassett, and Govindan (2015) assess path lengths and other properties for paths between popular content providers and their clients. A collection of 4 datasets were used throughout the study namely: (i) iPlane traceroutes from PlanetLab nodes towards 154k BGP prefixes, (ii) aggregated query counts per /24 prefix (3.8M) towards a large CDN, (iii) traceroute measurements towards 3.8M + 154k prefixes from Google's Compute Engine (GCE), Amazon Elastic Cloud, and IBM's Softlayer VMs, and (iv) traceroutes from RIPE Atlas probes towards cloud VMs and a number of popular websites. Using traceroute measurements from various platforms and converting the obtained IP hop path to its corresponding AS-level path the authors assess the network distance between popular content providers and client prefixes. iPlane traceroutes are used as a baseline for comparison, only 2% of these paths are one hop away from their destination this value increases to 40% (60%) for paths between GCE and iPlane (end user prefixes). This indicates that Google peers directly with the majority of networks which host its clients. Using the CDN logs as a proxy measure for traffic volume the authors find that Google peers with the majority of ASes which carry large volumes of traffic. Furthermore Chiu et al. find that the path from clients towards *google.com* due to off-net hosted cache servers is much shorter where 73% of queries come from ASes that either peer with Google or have an off-net server in their network or their upstream provider. A similar analysis for Amazon's EC2 and IBM's Softlayer was performed each having 30% and 40% one hop paths accordingly.

Kotronis, Nomikos, Manassakis, Mavrommatis, and Dimitropoulos (2017) study the possibility of improving latency performance through the employment of relay nodes within colocation facilities. This work tries to (i) identify the best locations/colos to place relay nodes and (ii) quantify the latency improvements that are attainable for end pairs. The authors select a set of ASes per each country which covers at least 10% of the countries population by using APNIC's IPv6 measurement campaign dataset APNIC (2018). RIPE Atlas nodes within these AS country pairs are selected which are running the latest firmware, are connected and pingable, and have had stable connectivity during the last 30 days. Colo relays are selected by relying on the set of pinned router interfaces from Giotsas, Smaragdakis, et al. (2015) work. Due to the age of the dataset, a series of validity tests including conformity with PeeringDB data, pingability, consistent ASN owner, and RTT-based geolocation test with Periscope LGs have been conducted over the dataset to filter out stale information. A set of PlanetLab relays and RIPE Atlas relays are also considered as reference points in addition to the set of colo relays. The measurement framework consists of 30 minute rounds between April 20th - May 17th 2017. Within each round, ping probes are sent between the selected end pairs to measure direct latency. Furthermore, the relay paths latency is estimated by measuring the latency between the <src, relay> and <dst, relay> pairs. The authors observed improve latency for 83% of cases with a median of 12-14ms between different relay types. Colo relays having the largest improvement. The number of required relays for improved latency is measured, the authors find that colo relays have the highest efficiency where 10 relays account for 58% of improved cases while the same number of improved cases for RIPE relays would require more than 100 relays. Lastly, the authors list the top 10 colo facilities which host the

20 most effective colo relays, 4 of these color are in the top 10 PeeringDB colos in terms of the number of colocated ASes and all host at least 2 or more IXPs within them.

Fontugne, Pelsser, Aben, and Bush (2017) introduce a statistical model for measuring and pin-pointing delay and forwarding anomalies from traceroute measurements. Given the prevalence of route asymmetry on the Internet, measuring the delay of two adjacent hops is not trivial. This issue is tackled by the key insight that differential delay between two adjacent hops is composed of two independent components. Changes in link latency can be detected by having a diverse set of traceroute paths that traverse the under study link and observing latency values disrupting the normal distribution for latency median. Forwarding patterns for each hop are established by measuring a vector accounting for the number of times a next hop address has been observed. Pearson product-moment correlation coefficient is used as a measure to detect deviations or anomalies within the forwarding pattern of a hop. RIPE Atlas' *built-in* and *anchoring* traceroute probes for an eight-month period in 2015 are used for the study. The authors highlight the applicability of their proposed methodology by providing insight into three historical events namely, DDoS attacks on DNS root servers, Telekom Malaysia's BGP route leak, and Amsterdam IXP outage.

2.4.1.3 Physical-Level Topology. Measuring characteristics of physical infrastructure using data plan measurements is very challenging due to the disassociation of routing from the physical layer. Despite these challenges, we overview two studies within this section that investigate the effects of sub-optimal fiber infrastructure on latency between two end-points Singla, Chandrasekaran,

Godfrey, and Maggs (2014) and attempt to measure and pinpoint the causes of observing subpar latency within fiber optic cables Bozkurt et al. (2018).

Singla et al. (2014) outline the underlying causes of sub-par latency within the Internet. The authors rely on about 400 Planet Lab nodes to periodically fetch the front page of popular websites, geolocate the webserver's location and measure the optimal latency based on speed of light (*c-latency*) constraints. Interestingly the authors find that the median of latency inflation is about 34 times greater than *c-latency*. Furthermore, the authors breakdown the webpage fetch time into its constituent components namely, DNS resolution, TCP handshake, and TCP transfer. Router path latency is calculated by conducting traceroutes towards the servers and lastly, minimum latency towards the web server is measured by conducting periodic ping probe. It is found that the median of router paths experience about 2.3x latency inflation. The authors hypothesize that latencies within the physical layer are due to sub-optimal fiber paths between routers. The validity of this hypothesis is demonstrated by measuring the pairwise distance between all nodes of Internet2 and GEANT network topologies and also computing road distance using Google Maps API. It is found that fiber links are typically 1.5-2x longer than road distances. While this inflation is smaller in comparison to webpage fetching component's latency the effects of fiber link inflation are evident within higher layers due to the stacked nature of networking layers.

Bozkurt et al. (2018) present a detailed analysis of the causes for sub-par latency within fiber networks. The authors rely on Durairajan et al. (2014) InterTubes dataset to estimate fiber lengths based on their conduits in the dataset. Using the infrastructure of a CDN, server clusters which are within a 25km radius of conduit endpoints were selected, and latency probes between pairs of servers at

both ends of the conduit were conducted every 3 hours for the length of 2 days. The conduit length is estimated using the speed of light within fiber optic cables (f-latency), and the authors find that only 11% of the links have RTTs within 25% of the f-latency for their corresponding conduit. Bozkurt et al. enumerate various factors which can contribute to the inflated latency that they observed within their measurements namely, (i) refraction index for different fiber optic cables varies, (ii) slack loops within conduits to account for fiber cuts, (iii) latency within optoelectrical and optical amplifier equipment, (iv) extra fiber spools to compensate for chromatic dispersion, (v) publication of mock routes by network operators to hide competitive details, and (vi) added fiber to increase latency for price differentiation. Using published latency measurements from AT&T and CenturyLink RTT inflation in comparison to f-latency from InterTubes dataset is measured to have a median of 1.5x (2x) for AT&T and CenturyLink's networks. The accuracy of InterTubes dataset is verified for Zayo's network. Zayo published detailed fiber routes on their website. The authors find great conformity for the majority of fiber conduit lengths while for 12% of links the length difference is more than 100km.

2.4.2 Resiliency. Studying the resiliency of Internet infrastructure has been the subject of many types of research over the past decade. While many of these studies have reported postmortems regarding natural disasters and their effects on Internet connectivity, others have focused on simulating what-if scenarios to examine the resiliency of the Internet towards various types of disruptions. Within these studies, researchers have utilized Internet topologies which were contemporary to their time. The resolution of these topologies would vary in accordance with the stated problem. For example, the resiliency of long haul fiber

infrastructure to rising sea levels due to global warming is measured by relying on physical topology maps Durairajan, Barford, and Barford (2018) while the effects of router outages on BGP paths and AS reachability is studied using a combination of router and AS level topologies within Luckie and Beverly (2017). The remainder of this section is organized according to the resolution of the underlying topology which is used by these studies.

2.4.2.1 AS-Level Topology.

Katz-Bassett et al. (2012) propose LIFEGUARD as a system for recovering from outages by rerouting traffic. Outages are categorized into two groups of forward and reverse path outages. Outages are detected and pinpointed by conducting periodic ping and traceroute measurements towards the routers along the path. A historical list of responsive routers for each destination is maintained. Prolonged unresponsive ping probes are attributed to outages. For forward path outages, the authors suggest the use of alternative upstream providers which traverse AS-paths that do not overlap with the unresponsive router. For reverse path outages, the authors propose a BGP poisoning solution where the origin AS would announce a path towards its own prefix which includes the faulty AS within the advertised path. This, in turn, causes the faulty AS to withdraw the advertisement (to avoid a loop) of the prefix and therefore cause alternative routes to be explored in the reverse path. A less-specific sentinel prefix is advertised by LIFEGUARD to detect the recovery of the previous path.

Luckie and Beverly (2017) correlate BGP outage events to inferred router outages by relying on time-series of IPID values obtained through active measurements. This work is motivated by the fact that certain routers rely on central incremental counters for the generated IPID values, given this assumption

one would expect to observe increasing IPID values for a single router. Any disruption in this pattern can be linked to a router reboot. IPID values for IPv4 packets are susceptible to counter rollover since they are only 16 bits wide. The authors rely on IPID values obtained by inducing fragmentation within IPv6 packets. The authors rely on a hit list of IPv6 router addresses which is obtained from intermediate hops of CAIDA's Ark traceroute measurements. By analyzing the time series of IPID values, an outage window is defined for each router. Router outages are correlated with their corresponding BGP control plane events by looking at BGP feeds and finding withdrawal and announcement messages occurring during the same time frame. It is found that for about 50% of router/prefix pairs at least 1-2 peers withdrew the prefix and nearly all peers withdrew their prefix announcement for about 10% of the router/prefix pairs. Luckie et al. find that about half of the ASes which had outages were completely unrouted during the outage period and had single points of failure.

Unlike Luckie et al. approach which relied on empirical data to assess the resiliency of Internet, Lad, Oliveira, Zhang, and Zhang (2007) investigate both the impact and resiliency of various ASes to prefix hijacking attacks by simulating different attacks using AS-level topologies obtained through BGP streams. Impact of prefix hijacking is measured as the fraction of ASes which believe the false advertisement by a malicious AS. Similarly, the resiliency of an AS against prefix hijacks is measured as the number of ASes which believe the true prefix origin announcement. Surprisingly it is found that 50% of stub and transit ASes are more resilient than Tier-1 ASes this is mainly attributed to valley-free route preferences.

Fontugne, Shah, and Aben (2018) look into structural properties, more specifically AS centrality, of AS-level IPv4 and IPv6 topology graphs. AS-level

topologies are constructed using BGP feeds of Routeviews, RIPE RIS, and BGPmon monitors. The authors illustrate the sampling bias of betweenness centrality (BC) measure by sub-sampling the set of available monitors and measuring the variation of BC for each sample. AS hegemony is used as an alternative metric for measuring the centrality of ASes which accounts for monitor biases by eliminating monitors too close or far from the AS in question and averaging the BC score across all valid monitors. Additionally, BC is normalized to account for the size of advertised prefixes. The AS hegemony score is measured for the AS-level graphs starting from 2004 till 2017. The authors find a great decrease in the hegemony score throughout the years supporting Internet flattening reports. Despite these observations, the hegemony score for ASes with the largest scores have remained consistent throughout the years pointing to the importance of large transit ASes in the operation of the Internet. AS hegemony for Akamai and Google is measured, the authors report little to no dependence for these content providers to any specific upstream provider.

*2.4.2.2 **Router-Level Topology.***

Palmer, Siganos, Faloutsos, Faloutsos, and Gibbons (2001) rely on topology graphs gathered by *SCAN* and *Lucent* projects consisting of 285k (430k) nodes (links) to simulate the effects of link and node failures within the Internet connectivity graph. The number of reachable pairs is used as a proxy measure to assess the impact of link or node failures. It is found that the number of reachable nodes does not vary significantly up to the removal of 50k links failures while this value drops to about 10k for node removals.

Kang and Gligor (2014); Kang, Lee, and Gligor (2013) propose the *Crossfire* denial of service attack that targets links which are critical for Internet connectivity

of ASes, cities, regions, or countries. The authors rely on a series of traceroute measurements towards addresses within the target entity and construct topological maps from various VPs towards these targets. The attacker would choose links that are “close" to the target (3-4 router hops) and appear with a high frequency within all paths. The attacker could cut these entities from the Internet by utilizing a bot-net to launch coordinated low rate requests towards various destinations in the target entity. Furthermore, the attacker can avoid detection by the target by targeting addresses which are in close proximity of the target entity, e.g. sending probes towards addresses within the same city where an AS resides within. The pervasiveness and applicability of the *Crossfire* attack is investigated by relying on 250 PlanetLab Chun et al. (2003) nodes to conduct traceroutes towards 1k web servers located within 15 target countries and cities. Links are ranked according to their occurrence within traceroutes and for all target cities and countries, the authors observe a very skewed power-law distribution. This observation is attributed to cost minimization within Internet routing (shortest path for intra-domain and hot-potato for inter-domain routing). Bottleneck links are measured to be on average about 7.9 (1.84) router (AS) hops away from the target.

Giotsas et al. (2017) develop *Kepler* a system that is able to detect peering outages. *Kepler* relies on BGP communities values that have geocoded embeddings. Although BGP community values are not standardized, they have been utilized by ASes for traffic engineering, traffic blackholing, and network troubleshooting. Certain ASes use the lower 16bits of the BGP communities attribute as a unique identifier for each of their border routers. These encodings are typically documented on RIR webpages. The authors compile a dictionary of BGP community values and their corresponding physical location (colo or IXP) by

parsing RIR entries. Furthermore, a baseline of stable BGP paths is established by monitoring BGP feeds and removing transient announcements. Lastly, the tenants of colo facilities and available IXPs and their members is compiled from PeeringDB, DataCenterMap, and individual ASes websites. Deviations in stable BGP paths such as explicit withdrawal or change in BGP community values are considered as outage signals.

2.4.2.3 Physical-Level Topology. Schulman and Spring (2011) investigate outages within the last mile of Internet connectivity which are caused by severe weather conditions. The authors design a tool called *ThunderPing* which relies on weather alerts from the US National Weather Service to conduct connectivity probes prior, during, and after a severe weather condition towards the residential users of the affected regions. A list of residential IP addresses is compiled by parsing the reverse DNS entry for 3 IP addresses within each /24 prefix. If any of the addresses have a known residential ISP such as Comcast or Verizon within their name the remainder of addresses within that block are analyzed as well. IP addresses are mapped to their corresponding geolocation by relying on Maxmind's IP to GEO dataset. Upon the emergence of a weather alert *ThunderPing* would ping residential IP addresses within the affected region for 6 hours before, during, and after the forecasted event using 10 geographically distributed PlanetLab nodes. A sliding window containing 3 pings is used to determine the state of a host. A host responding with more than half of the pings is considered to be *UP*, not responding to any pings is considered to be *DOWN*, and host responding to less than half of the pings is in a *HOSED* state. The authors find that failure rates are more than double during thunderstorms compared to other weather conditions. Furthermore, the median for the duration

of *DOWN* times is almost an order of magnitude larger (10^4 seconds) during thunderstorms compared to clear weather conditions.

Eriksson, Durairajan, and Barford (2013) present a framework (*RiskRoute*) for measuring the risks associated with various Internet routes. *RiskRoute* has two main objectives namely, (i) computing backup routes and (ii) to measure new paths for network provisioning. The authors introduce the *bit-risk miles* measure which quantifies the geographic distance that is traveled by traffic in addition to the outage risk along the path both in historical and immediate terms. Furthermore *bit-risk miles* is scaled to account for the impact of an outage by considering the population that is in the proximity of an outage. The likelihood of historical outage for a specific location is estimated using a Gaussian kernel which relies on observed disaster events at all locations. For two PoPs, RiskRoute aims to calculate the path which minimizes the *bit-risk mile* measure. For intra-domain routes, this is simply calculated as the path which minimizes the *bit-risk mile* measure among all possible paths which connect the two PoPs. For inter-domain routing the authors estimate BGP decisions using geographic proximity and rely on shortest path routes. Using the RiskRoute framework, improvements in the robustness of networks is analyzed by finding an edge which would result in the largest increase in *bit-risk* measure among all possible paths. It is found that Sprint and Teliasonera networks observe the greatest improvement in robustness while Level3's robustness remains fairly consistent mostly due to rich connectivity within its network.

Durairajan et al. (2018) assess the impact of rising sea levels on the Internet infrastructure within the US. The authors align the data from the sea level rise inundation dataset from the National Oceanic and Atmospheric Administration (NOAA) with long-haul fiber maps from the Internet Atlas project Durairajan et

al. (2013) using the *overlap* feature of ArcGIS. The amount of affected fibers as well as the number of PoPs, colos, and IXPs that will be at risk due to the rising sea levels is measured. The authors find that New York, Seattle, and Miami are among the cities with the highest amount of vulnerable infrastructure.

2.4.3 AS Relationship Inference. ASes form inter-AS connections motivated by different business relationships. These relationships can be in the form of a transit AS providing connectivity to a smaller network as a customer (c2p) by charging them based on the provided bandwidth or as a settlement-free connection between both peers (p2p) where both peers exchange equal amounts of traffic through their inter-AS link. These inter-AS connections are identical from topologies obtained from control or data plan measurements. The studies within this section overview a series of methodologies developed based on these business relationships in conjunction with the valley free routing principle to distinguish these peering relationships from each other.

2.4.3.1 AS-Level. Luckie, Huffaker, Dhamdhere, and Giotsas (2013) develop an algorithm for inferring the business relationships between ASes by solely relying on BGP data. Relationships are categorized as a customer to provider (c2p) relationship were a customer AS pays a provider AS for its connectivity to the Internet or a peer to peer (p2p) relationship were two ASes provide connectivity to each other and often transmit equal amounts of traffic through their inter-AS link(s). Inference of these relationships are based on BGP data using three assumptions: (i) there is a clique of large transit providers at the top of the Internet hierarchy, (ii) customers enter a transit agreement to be globally reachable, and (iii) we shouldn't have a cycle in customer to provider (c2p) relationships. The authors validate a subset (43k) of their inferences, which is the largest by the

time of publication, and finally they provide a new solution for inferring customer cones of ASes. For their analyses, the authors rely on various data sources namely BGP paths from Routeviews and RIPE's RIS, any path containing origin ASes which do not contain valid ASNs (based on RIRs) is excluded from the dataset. For validation Luckie et al. use three data sources: validation data reported by network operators to their website, routing policies reported to RIRs in *export* and *import* fields, and finally they use the communities attribute of BGP announcements based on the work of Giotsas and Zhou (2012). The authors define two metrics node degree and transit degree which can be measured from the AS relationship graph.

Giotsas, Luckie, et al. (2015) modify CAIDA's IPv4 relationship inference algorithm Luckie, Huffaker, Dhamdhere, and Giotsas (2013) and adapt it to IPv6 networks with the intention of addressing the lack of a fully-connected transit-free clique within IPv6 networks. BGP dumps from Routeviews and RIPE RIS which announce reachability towards IPv6 prefixes are used throughout this study. For validation of inferred relationships three sources are used: BGP communities, RPSL which is a route policy specification language that is available in WHOIS datasets and is mandated for IXPs within EU by RIPE, and local preference (LocPref) which is used to indicate route preference by an AS where ASes assign higher values to customers and lower values to providers to minimize transit cost. Data is sanitized by removing paths with artifacts such as loops or invalid ASNs. The remainder of the algorithm is identical to Luckie, Huffaker, Dhamdhere, and Giotsas (2013) with modifications to two steps: i) inferring the IPv6 clique and ii) removing c2p inferences made between stub and clique ASes. In addition to considering the transit degree and reachability, peering policy of ASes is also taken into account for identifying cliques. Peering policy is extracted from PeeringDB,

a restrictive policy is assumed for ASes who do not report this value. ASes with selective or restrictive policies are selected as seeds to the clique algorithm. For an AS to be part of the clique, it should provide BGP feeds to Routeviews or RIPE RIS and announce routes to at least 90% of IPv6 prefixes available in BGP. The accuracy of inferences is validated using the three validation sources which where described, a consistent accuracy of at least 96% was observed for p2c and p2p relationships for the duration of the study. The fraction of congruent relationships where the relationship type is identical for IPv4 and IPv6 networks is measured. The authors find that this fraction increases from 85% in 2006 to 95% in 2014.

2.4.3.2 ***PoP-Level.*** Giotsas et al. (2014) provide a methodology for extending traditional AS relationship models to include two complex relationships namely: hybrid and partial transit relationships. Hybrid relations indicate different peering relations at different locations. Partial transit relations restrict the scope of a customers relation by not exporting all provider paths to the customer. AS path, prefixes, and communities strings are gathered from Routeviews and RIPE RIS datasets. CAIDA's Ark traceroutes in addition to a series of targeted traceroutes launched from various looking glasses are employed to confirm the existence of various AS relationships. Finally, geoinformation for AS-links are gathered from BGP community information, PeeringDB's reverse DNS scan of IXP prefixes, DNS parsing of hostnames by CAIDA's DRoP service, and NetAcuity's IP geolocation dataset is used as a fallback when other methods do not return a result. Each AS relationship is labeled into one of the following export policies: i) full transit (FT) where the provider exports prefixes from its provider, ii) partial transit (PT) where prefixes of peers and customers are only exported, and iii) peering (P) where prefixes of customers are only exported. Each identified relationship defaults to

peering unless counter facts are found through traceroute measurements which indicate PT or FT relationships. Out of 90k p2c relationships 4k of them are classified as complex with 1k and 3k being hybrid and partial-transit accordingly. For validation (i) direct feedback from network operators, (ii) parsed BGP community values, and (iii) RPSL objects are used. Overall 19% (7%) of hybrid (partial-transit) relationships were confirmed.

CHAPTER III

LOCALITY OF TRAFFIC

This chapter provides a study on the share of cloud providers and CDNs in Internet traffic from the perspective of an edge network (UOnet). Furthermore, this work quantifies the degree to which the serving infrastructure for cloud providers and CDNs is close/local to UOnet's network and investigates the implications of this proximity on end-users performance.

The content in this chapter is derived entirely from Yeganeh, Rejaie, and Willinger (2017) as a result of collaboration with co-authors listed in the manuscript. Bahador Yeganeh is the primary author of this work and responsible for conducting all the presented analyses.

3.1 Introduction

During the past two decades, various efforts among different Internet players such as large Internet service providers (ISP), commercial content distribution networks (CDN) and major content providers have focused on supporting the *localization* of Internet traffic. Improving traffic localization has been argued to ensure better user experience (in terms of shorter delays and higher throughput) and also results in less traffic traversing an ISP's backbone or the interconnections (i.e., peering links) between the involved parties (e.g., eyeball ASes, transit providers, CDNs, content providers). As a result, it typically lowers a network operator's cost and also improves the scalability of the deployed infrastructure in both the operator's own network and the Internet at large.

The main idea behind traffic localization is to satisfy a user request for a certain piece of content by re-directing the request to a cache or front-end server that is in close proximity to that user and can serve the desired piece

of content. However, different commercial content distribution companies use different strategies and deploy different types of infrastructures to implement their business model for getting content closer to the end users. For example, while Akamai Akamai (2017) operates and maintains a global infrastructure consisting of more then 200K servers located in more than 1.5K different ASes to bring the requested content by its customers closer to the edge of the network where this content is consumed, other CDNs such as Limelight or EdgeCast rely on existing infrastructure in the form of large IXPs to achieve this task Limelight (2017). Similar to Akamai but smaller in scale, major content providers such as Google and Netflix negotiate with third-party networks to deploy their own caches or servers that are then used to serve exclusively the content provider's own content. In fact, traffic localization efforts in today's Internet continue as the large cloud providers (e.g., Amazon, Microsoft) are in the process of boosting their presence at the edge of the network by deploying increasingly in the newly emerging 2nd-tier datacenters (e.g., EdgeConneX EdgeConneX (2018)) that target the smaller- or medium-sized cities in the US instead of the major metropolitan areas.

These continued efforts by an increasing number of interested parties to implement ever more effective techniques and deploy increasingly more complex infrastructures to support traffic localization has motivated numerous studies on designing new methods and evaluating existing infrastructures to localize Internet traffic. While some of these studies Adhikari et al. (2012); Böttger, Cuadrado, Tyson, Castro, and Uhlig (2016); Calder et al. (2013); Fan, Katz-Bassett, and Heidemann (2015) have focused on measurement-based assessments of different deployed CDNs to reveal their global Böttger et al. (2016); Calder et al. (2013) or local Gehlen, Finamore, Mellia, and Munafò (2012); Torres et al. (2011)

infrastructure nodes, others have addressed the problems of reverse-engineering a CDN's strategy for mapping users to their close-by servers or examining whether or not the implemented re-direction techniques achieve the desired performance improvements for the targeted end users Adhikari et al. (2012); Fan et al. (2015); Gehlen et al. (2012). However, to our knowledge, none of the existing studies provides a detailed empirical assessment of the nature and impact of traffic localization as seen from the perspective of an actual stub-AS. In particular, the existing literature on the topic of traffic localization provides little or no information about the makeup of the content that the users of an actual stub-AS request on a daily basis, the proximity of servers that serve the content requested by these users (overall or per major content provider), and the actual performance benefits that traffic localization entails for the consumers of this content (i.e., end users inside the stub-AS).

In this chapter, we fill this gap in the existing literature and report on a measurement study that provides a detailed assessment of different aspects of the content that arrives at an actual stub-AS as a result of the requests made by its end users. To this end, we consider multiple daily snapshots of unsampled Netflow data for all exchanged traffic between a stub-AS that represents a Research & Education network (i.e., UOnet operated by the University of Oregon) and the Internet 3.2. We show that some 20 content providers are responsible for most of the delivered traffic to UOnet and that for each of these 20 content providers, the content provider specific traffic is typically coming from only a small fraction of source IPs (Section 3.3). Using RTT to measure the distance of these individual source IPs from UOnet, we present a characterization of this stub-AS' *traffic footprint*; that is, empirical findings about the locality properties of delivered

traffic to UOnet, both in aggregate and at the level of individual content providers (Section 3.4). In particular, we examine how effective the individual content providers are in utilizing their infrastructure nodes to localize their delivered traffic to UOnet and discuss the role that *guest servers* (i.e., front-end servers or caches that some of these content providers deploy in third-party networks) play in localizing traffic for this stub-AS (Section 3.5). As part of this effort, we focus on Akamai and develop a technique that uses our data to identify all of Akamai's guest servers that delivered content to UOnet. We then examine different features of the content that arrived at UOnet from those guest servers as compared to the content that reached UOnet via servers located in Akamai's own AS. Finally, we investigate whether or not a content provider's ability to localize its traffic has implications on end user-perceived performance, especially in terms of observed throughput (Section 3.6).

3.2 Data Collection for a Stub-AS: UOnet

The stub-AS that we consider for this study is the campus network of the University of Oregon (UO), called UOnet (ASN3582). UOnet serves more than 24K (international and domestic) students and 4.5K faculty/staff during the academic year. These users can access the Internet through UOnet using wireless (through 2000+ access points) or wired connections. Furthermore, more than 4,400 of the students reside on campus and can access the Internet through UOnet using their residential connections. UOnet has three upstream providers, Neronet (AS3701), Oregon Gigapop (AS4600) and the Oregon IX exchange. Given the types of offered connectivity and the large size and diversity of the UOnet user population, we consider the daily traffic that is delivered from the rest of the Internet to UOnet

to be representative of the traffic that a stub-AS that is classified as a US Research & Education network is likely to experience.

To conduct our analysis, we rely on un-sampled Netflow (v5) data that is captured at the different campus border routers. As a result, our Netflow data contains all of the flows between UOnet users and the Internet. The Netflow dataset contains a separate record for each incoming (and outgoing) flow from (to) an IP address outside of UOnet, and each record includes the following flow attributes: *(i)* source and destination IP addresses, *(ii)* source and destination port numbers, *(iii)* start and end timestamps, *(iv)* IP protocol, *(v)* number of packets, and *(vi)* number of bytes. We leverage Routeviews data to map all the external IPs to their corresponding Autonomous Systems (ASes) and use this information to map individual flows to particular providers (based on their AS number) and then determine the number of incoming (and outgoing) flows (and corresponding bytes) associated with each provider. In our analysis, we only consider the incoming flows since we are primarily interested in delivered content and services from major content providers to UOnet users. An incoming flow refers to a flow with the source IP outside and destination IP inside UOnet. We select 10 daily (24 hour) snapshots of Netflow data that consist of Tuesday and Wednesday from five consecutive weeks when the university was in session, starting with the week of Oct 3rd and ending with the week of of Oct 31st in 2016. Table 2 summarizes the main features of the selected snapshots, namely their date, the number of incoming flows and associated bytes, and the number of unique external ASes and unique external IPs that exchanged traffic with UOnet during the given snapshot. In each daily snapshot, wireless connections are responsible for roughly 62% (25%) of delivered

Table 2. Main features of the selected daily snapshots of our UOnet Netflow data.

Snapshot	Flows (M)	TBytes	ASes (K)	IPs (M)
10/04/16	196	8.7	39	3.3
10/05/16	193	8.5	37	3.0
10/11/16	199	9.0	41	4.1
10/12/16	198	9.1	41	4.7
10/18/16	202	8.8	40	3.7
10/19/16	200	9.1	38	3.3
10/25/16	205	8.7	37	2.9
10/26/16	209	9.1	40	4.1
11/01/16	212	8.6	39	3.5
11/02/16	210	8.7	40	4.3

bytes (flows) and residential users contributed to about 17% (10%) of incoming bytes (flows).

3.3 Identifying Major Content Providers

Our main objective is to leverage the UOnet dataset to provide an empirical assessment of traffic locality for delivered flows to UOnet and examine its implications for the end users served by UOnet. Here by “locality" we refer to a notion of network distance between the servers in the larger Internet that provide the content/service requested from within UOnet. Since the level of locality of delivered traffic by each content provider depends on both the relative network distance of its infrastructure and its strategy for utilizing this infrastructure, we conduct our analysis at the granularity of individual content providers and focus only on those that are responsible for the bulk of delivered content to UOnet. Moreover, because the number of unique source IPs that send traffic to UOnet on a daily basis is prohibitively large, we identify and focus only on those IPs that are responsible for a significant fraction of the delivered traffic.

Inferring Top Content Providers: Figure 6 (left y-axis) shows the histogram of delivered traffic (in TB) to UOnet by those content providers that have the largest

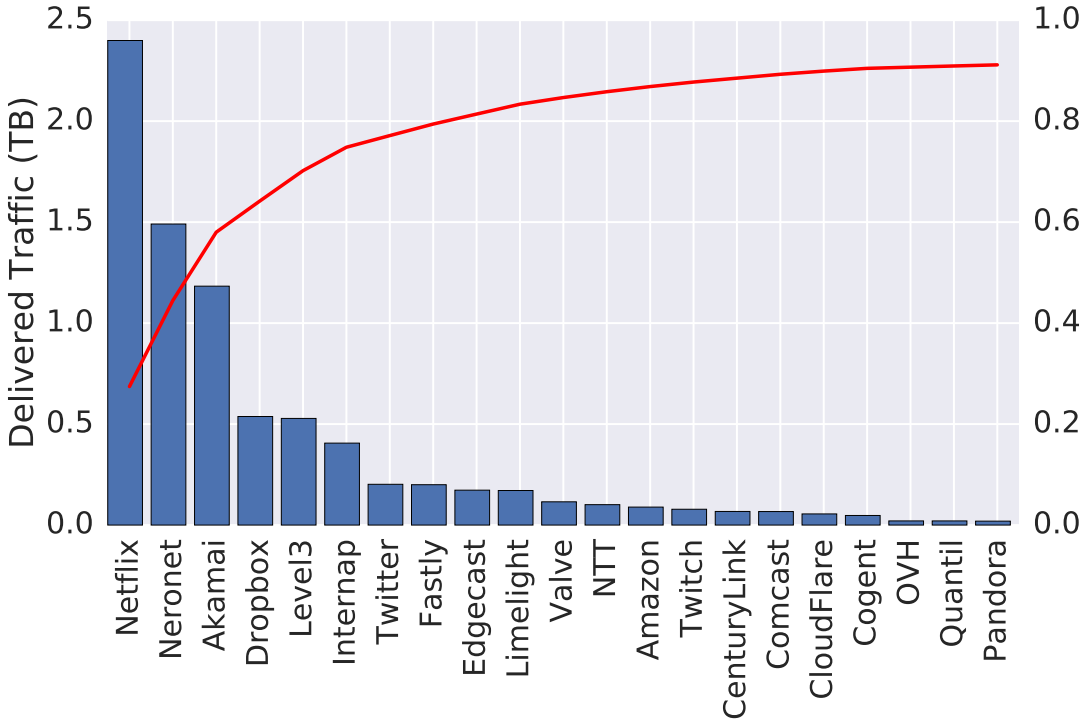


Figure 6. The volume of delivered traffic from individual top content providers to UOnet along with the CDF of aggregate fraction of traffic by top 21 content providers in the 10/04/16 snapshot.

contributions in the 10/04/16 snapshot. It also shows (right y-axis) the CDF of the fraction of aggregate traffic that is delivered by the top-k content providers in this snapshot. The figure is in full agreement with earlier studies such as Ager, Mühlbauer, Smaragdakis, and Uhlig (2011); Chatzis et al. (2013) and clearly illustrates the extreme skewness of this distribution – the top 21 content providers (out of some 39K ASes) are responsible for 90% of all the delivered daily traffic to UOnet.

To examine the stability of these top content providers across our 10 daily snapshots, along the x-axis of Figure 7, we list any content provider that is among the top content providers (with 90% aggregate contributions in delivered traffic) in at least one daily snapshot (the ordering is in terms of mean rank, from small to

large for content providers with same prevalence). This figure shows the number of daily snapshots in which a content provider has been among the top content providers (i.e. content provider's prevalence, left y-axis) along with the summary distribution (i.e., box plot) of each of the content providers rankings among the top content providers across different snapshots (rank distribution, right y-axis). We observe that the same 21 content providers consistently appear among the top content providers. These 21 content providers are among the well-recognized players of today's Internet and include major content providers (e.g. Netflix, Twitter), widely-used CDNs (e.g. Akamai, LimeLight and EdgeCast), and large providers that offer hosting, Internet access, and cloud services (e.g. Comcast, Level3, CenturyLink, Amazon). In the following, we only focus on these 21 content providers (called *target content providers*) that are consistently among the top content providers in all of our snapshots. These target content providers are also listed in Figure 6 and collectively contribute about 90% of the incoming daily bytes in each of our snapshots.

Inferring Top IPs per Target Content Providers: To assess the locality of the traffic delivered to UOnet from each target content provider, we consider the source IP addresses for all of the incoming flows in each daily snapshot. While for some target content providers, the number of unique source IP addresses is as high as a few tens of thousands, the distribution of delivered traffic across these IPs exhibits again a high degree of skewness; i.e. for each target content provider, only a small fraction of source IPs (called *top IPs*) is responsible for 90% of delivered traffic. Figure 8 shows the summary distribution (in the form of box plots) of the number of top IPs across different snapshots along with the cumulative number of unique top IPs (blue line) and all IPs (red line) across all

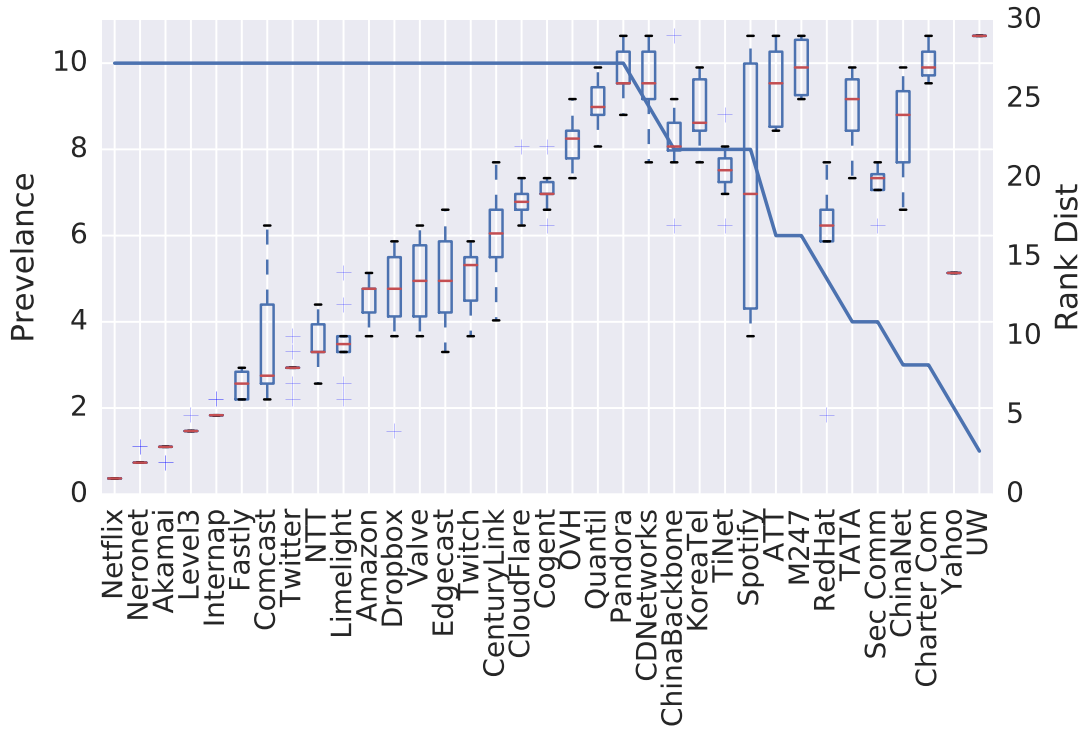


Figure 7. The prevalence and distribution of rank for any content provider that has appeared among the top content providers in at least one daily snapshot.

of our 10 snapshots. The log-scale on the y-axis shows that the number of top IPs is often significantly smaller than the number of all IP addresses (as a result of the skewed distribution of delivered content by different IPs per target content provider). A small gap between the total number of top IPs and their distribution across different snapshots illustrates that for many of the target content providers, the top IPs do not vary widely across different snapshots. In our analysis of traffic locality below, we only consider the collection of all top IPs associated with each of the target content provider across different snapshots. Focusing on these roughly 50K IPs allows us to capture a rather complete view of delivered traffic to UOnet without considering the millions of observed source IPs.

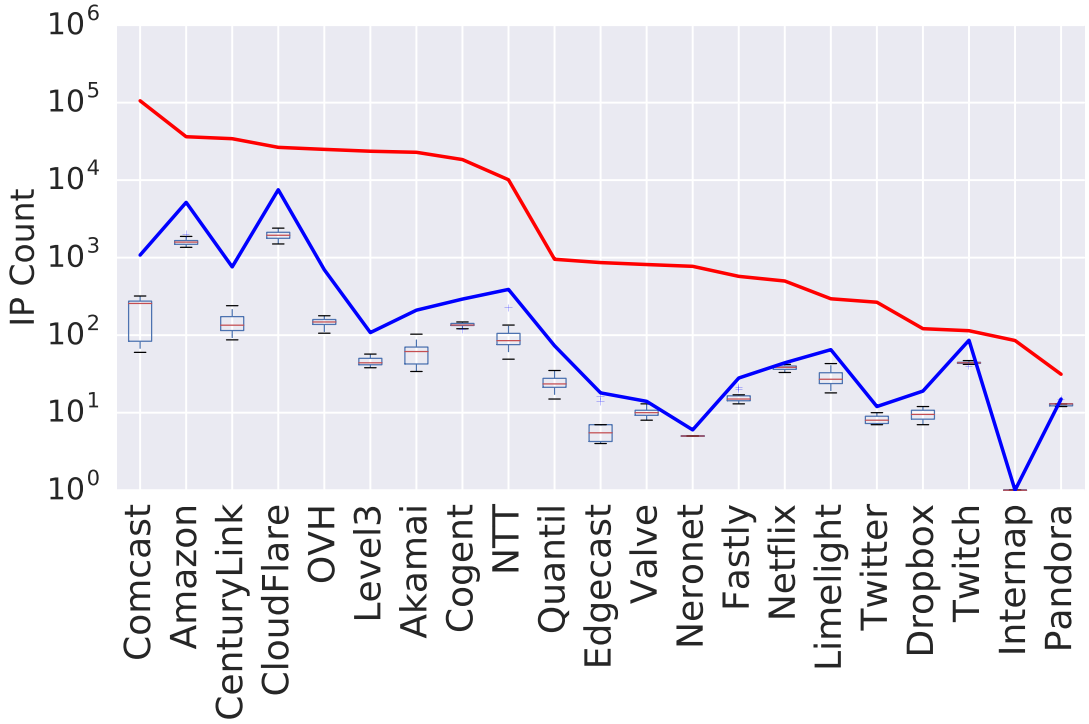


Figure 8. Distribution of the number of top IPs across different snapshots in addition to total number of unique top IP addresses (blue line) and the total number of unique IPs across all snapshots (red line) for each target content provider.

Measuring the Distance of Top IPs: Using the approximately 50K top IPs for all 21 target content providers, we conducted a measurement campaign (on 11/10/16) that consisted of launching 10 rounds of traceroutes[1] from UOnet to all of these 50K top IPs to infer their minimum RTT.

Note that the value of RTT for each top IP accounts for possible path asymmetry between the launching location and the target IP and is therefor largely insensitive to the direction of the traceroute probe (i.e. from UOnet to a top IP vs. from a top IP to UOnet). Our traceroute probes successfully reached 81% of the targeted IP addresses. We exclude three target content providers (i.e., Internap,

[1]We use all three types of traceroute probes(TCP, UDP, ICMP) and spread them throughout the day to reach most IPs and reliable capture minimum RTT

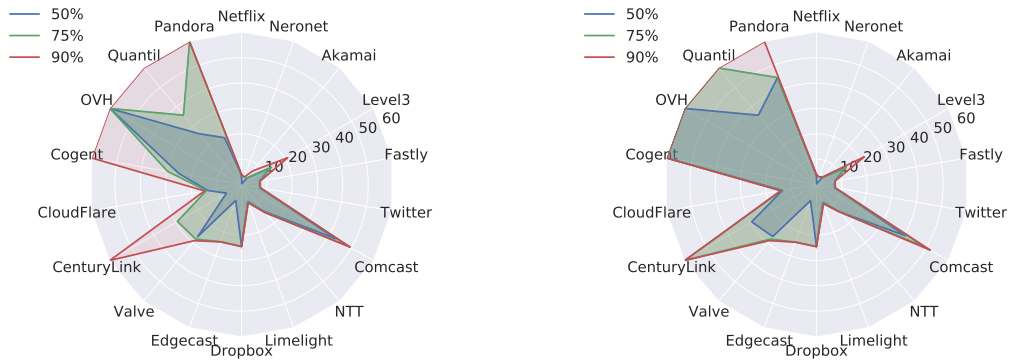


Figure 9. Radar plots showing the aggregate view of locality based on RTT of delivered traffic in terms of bytes (left plot) and flows (right plot) to UOnet in a daily snapshot (10/04/2016).

Amazon and Twitch) from our analysis because their servers did not respond to more than 90% of our traceroute probes. All other target content providers responded to more than 90% of our probes.

The outcome of our measurement campaign is the list of top IPs along with their min RTT and the percentage of delivered traffic (in terms of bytes and flows) for each target content provider. With the help of this information, we can now assess the locality properties of the content that is delivered from each target content provider to UOnet. Note that in theory, any distance measure could be used for this purpose. However, in practice, neither AS distance (i.e., number of AS hops), nor hop-count distance (i.e., number of traceroute hops), nor geographic distance are reliable metrics. While the first two ignore the commonly encountered asymmetry of IP-level routes in today's Internet Sánchez et al. (2013), the last metric suffers from known inaccuracies in commercial databases such as IP2Location IP2Location (2015) and Maxmind MaxMind (2018) that are commonly used for IP geolocation. We choose the RTT distance (i.e., measured by min RTT value) as our metric-of-choice for assessing the locality of delivered

traffic since it is the most reliable distance measure and also the most relevant in terms of user-perceived delay.

3.4 Traffic Locality for Content Providers

Overall View of Traffic Locality: We use radar plots to present an overall view of the locality of aggregate delivered traffic from our target content providers to UOnet based on RTT distance. Radar plots are well suited for displaying multi-variable data where individual variables are shown as a sequence of equiangular spokes, called radii. We use each spoke to represent the locality of traffic for a given target content provider by showing the RTT values for 50th, 75th and 90th percentiles of delivered traffic (in bytes or flows). In essence, the spoke corresponding to a particular target content provider shows what percentage of the traffic that this content provider delivers to UOnet originates from within 10, 20,..., or 60ms distance from our stub-AS. Figure 9 shows two such radar plots for a single daily snapshot (10/04/16). In these plots, the target CPs are placed around the plot in a clock-wise order (starting from 12 o' clock) based on their relative contributions in delivered bytes (as shown in Figure 6), and the distances (in terms of min RTT ranges) are marked on the 45-degree spoke. The left and right plots in Figure 9 show the RTT distance for 50, 75 and 90th percentile of delivered bytes and flows for each content provider, respectively. By connecting the same percentile points on the spokes associated with the different target content providers, we obtain a closed contour where the sources for 50, 75 or 90% of the delivered content form our target content providers to UOnet are located. We refer to this collection of contours as the *traffic footprint* of UOnet. While more centrally-situated contours indicate a high degree of overall traffic locality for the

considered stub-AS, contours that are close to the radar plot's boundary for some spokes suggest poor localization properties for some content providers.

The radar plots in Figure 9 show that while there are variations in traffic locality for different target content providers, 90% of the delivered traffic for the top 13 content providers are delivered from within a 60ms RTT distance from UOnet and for 9 of them from within 20ms RTT. Moreover, considering the case of Cogent, while 50% of bytes from Cogent are delivered from an RTT distance of 20ms, 50% of the flows are delivered from a distance of 60ms. Such an observed higher level of traffic locality with respect to bytes compared to flows suggests that a significant fraction of the corresponding target content provider's (in this case, Cogent) large or "elephant" flows are delivered from servers that are in closer proximity to UOnet than those that serve the target content provider's smaller flows. Collectively, these findings indicate that for our stub-AS, the overall level of traffic locality for delivered bytes and flows is high but varies among the different target content providers. These observations are by and large testimony to the success of past and ongoing efforts by the different involved parties to bring content closer to the edge of the network where it is requested and consumed. As such, the results are not surprising, but to our knowledge, they provide the first quantitative assessment of the per-content provider traffic footprint (based on RTT distance) of a stub-AS.

Variations in Traffic Locality: After providing an overall view of the locality of the delivered traffic to UOnet for a single snapshot, we next turn our attention to how traffic locality of a content provider (with respect to UOnet) varies over time. To simplify our analysis, we consider all flows of each target content provider and bin them based on their RTTs using a bin size of 2ms. The flows in each bin

are considered as a single group with an RTT value given by the mid-bin RTT value. We construct the histogram of percentages of delivered bytes from each group of flows in each bin and define the notion of *Normalized Weighted Locality* for delivered traffic from a provider P in snapshot s as:

$$NWL(s,P) = \sum_{i \epsilon RTTBins(P)} \frac{FracBytes(i) * RTT(i)}{minRTT(P)}$$

NWL(s,P) is simply the sum of the fraction of delivered traffic from each RTT bin ($FracBytes(i)$) that is weighted by its RTT and then normalized by the lowest RTT among all bin ($minRTT(P)$) for a content provider across all snapshots. NWL is an aggregate measure that illustrates how effectively a content provider localizes its delivered traffic over its own infrastructure. A NWL value of 1 implies that all of the traffic is delivered from the closest servers while larger values indicate more contribution from servers that are further from UOnet.

The top plot in Figure 10 presents the summary distribution of $NWL(s,P)$ across different daily snapshots for each content provider. The bottom plot in Figure 10 depicts min RTT for each content provider. These two plots together show how local the closest server of a content provider is and how effective each content provider is in utilizing its infrastructure. The plots also demonstrate the following points about the locality of traffic. For one, for many target content providers (e.g. Netflix, Comcast, Valve), the NWL values exhibits small or no variations across different snapshots. Such a behavior suggests that the pattern of delivery from different servers is stable across different snapshots. In contrast, for content providers with varying NWL values, the contribution of various servers (i.e. the pattern of content delivery from various content provider servers) changes over time. Second, the value of NWL is less than 2 (and often very close to 1) for many content providers. This in turn indicates that these content providers

effectively localize their delivered traffic to UOnet over their infrastructure. The value of NWL for other content providers is larger and often exhibit larger variation due to their inability to effectively utilize their nodes to localize delivered traffic to UOnet.

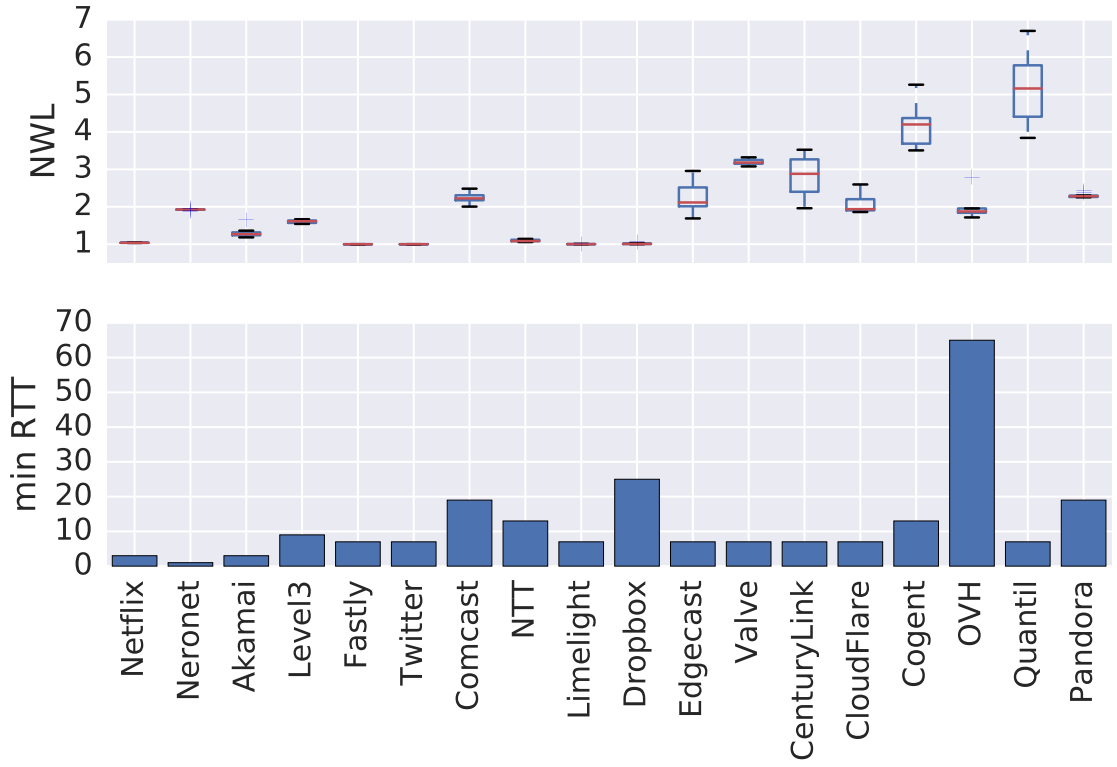


Figure 10. Two measures of traffic locality, from top to bottom, Summary distribution of NWL and the RTT of the closest servers per content provider (or minRTT).

3.5 Traffic From Guest Servers

To improve the locality properties of their delivered content and services to end users, some content providers expand their infrastructure by deploying some of their servers in other networks. We refer to such servers as *guest servers* and to the third-party networks hosting them as *host networks or host ASes*. For example, Akamai is known to operate some 200K such servers in over 1.5K different host

networks, with the servers using IP addresses that belong to the host networks Fan et al. (2015); Triukose, Wen, and Rabinovich (2011).

We present two examples to illustrate the deployment of guest servers. First, our close examination of delivered traffic from Neronet which is one of UOnet's upstream providers revealed that all of its flows are delivered from a small number of IPs (see Figure 8) associated with Google servers, i.e. Google caches Calder et al. (2013) that are deployed in Neronet. This implies that all of Google's traffic for UOnet is delivered from Neronet-based Google caches and explains why Google is not among our target content providers. Second, Netflix is known to deliver its content to end users through its own caches (called Open Connect AppliancesNetflix (2017b)) that are either deployed within different host networks or placed at critical IXPs Böttger et al. (2016). When examined the DNS names for all the source IPs of our target content providers, we observed a number of source IPs that are within another network and their DNS name follow the `*.pdx001.ix.nflxvideo.net` format. This is a known Neflix convention for DNS names and clearly indicates that these guest servers are located at an IXP in Portland, Oregon Böttger et al. (2016).

3.5.1 Detecting Guest Servers. Given the special nature of content delivery to UOnet from Google (via Neronet) and Netflix (via a close-by IXP), we focus on Akamai to examine how its use of guest servers impact the locality of delivered traffic to UOnet. However, since our basic methodology that relies on a commonly-used IP-to-AS mapping technique cannot identify Akamai's guest servers and simply associates them with their host network, we present in the following a new methodology for identifying Akamai's guest servers that deliver content to UOnet.

Our proposed method leverages Akamai-specific information and proceeds in two steps. The first step consists of identifying the URLs for a few small, static and popular objects that are likely to be cached at many Akamai servers. Then, in a second step, we probe the observed source IP addresses at other target content providers with properly-formed HTTP request for the identified objects. Any third-party server that provides the requested objects is considered an Akamai guest server. More precisely, we first identify a few Akamai customer websites and interact with them to identify small, static and popular objects (i.e., "reference objects"). Since JavaScript or CSS files are less likely to be modified compared to other types of objects and thus are more likely to be cached by Akamai servers, we used in our experiments two JavaScript objects and a logo from Akamai client web sites (e.g. Apple, census.gov, NBA). Since an Akamai server is responsible for hosting content from multiple domain names, the web server needs a way to distinguish requests that are redirected from clients of different customer websites. This differentiation is achieved with the help of the HOST field of the HTTP header. Specifically, when constructing a HTTP request to probe an IP address, we set the HOST field to the original domain name of the reference object (e.g. apple.com, census.gov, nba.com). Next, for each reference object, we send a separate HTTP request to each of the 50K top source IP addresses in our datasets (see Section 3). If we receive the HTTP OK/200 status code in response to our request and the first 100 bytes of the provided object match the requested reference object [2], we consider the server to be an Akamai guest server and identify its AS as host AS. We repeat our request using other reference objects if the HTTP request

[2]The second condition is necessary since some servers provide a positive response to any HTTP requests.

fails or times-out. If all of our requests time-out or receive a HTTP error code, we mark the IP address as a non-Akamai IP address.

To evaluate our proposed methodology, we consider all the 601 servers in our dataset whose IP addresses are mapped to Akamai (based on IP to AS mapping) and send our HTTP requests to all of them. Since all Akamai servers are expected to behave similarly, the success rate of our technique in identifying these Akamai servers demonstrates its accuracy. Indeed, we find that 585 (97%) of these servers properly respond to our request and are thus identified as Akamai servers. The remaining 3% either do not respond or respond with various HTTP error codes. When examining these 16 failed servers more closely, we discovered that 11 of them were running a mail server and would terminate a connection to their web server regardless of the requested content. This suggests that these Akamai servers perform functions other than serving web content.

Using our proposed technique, we probed all 50K top source IP addresses associated with our 21 target content providers in all of our snapshots. When performing this experiment (on 11/20/16), we discovered between 143-295 Akamai guest servers in 3-7 host ASes across the different snapshots. In total, there were 658 unique guest servers from 7 unique host ASes, namely NTT, CenturyLink, OVH, Cogent, Comcast, Dropbox and Amazon. Moreover, these identified Akamai guest servers deliver between 121-259 GBytes to UOnet in their corresponding daily snapshots which is between 9-20% of the aggregate daily traffic delivered from Akamai to UOnet. These results imply that the 34-103 Akamai-owned servers in each snapshot deliver on average 12 times more content to UOnet than Akamai's 143-295 guest servers. Moreover, we observed that the bulk of delivered bytes from Akamai's guest servers to UOnet (i.e., 98%) is associated with guest servers that

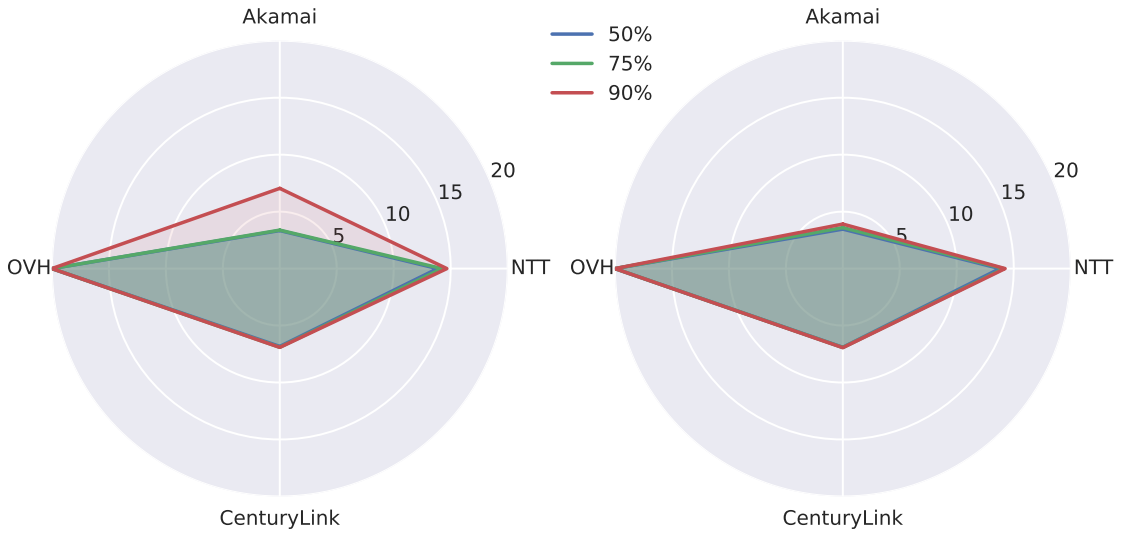


Figure 11. Locality (based on RTT in ms) of delivered traffic (bytes, left plot; flows, right plot) for Akamai-owned servers as well as Akamai guest servers residing within three target ASes for snapshot 2016-10-04.

are deployed in two content providers, namely NTT (76.1%) and CenturyLink (21.9%).

3.5.2 Relative Locality of Guest Servers. Deploying guest servers in various host ASes enables a content provider to either improve the locality of its traffic or provide better load balancing among its servers. To examine these two objectives, we compare the level of locality of traffic delivered from Akamai-owned servers vs Akamai's guest servers. The radar plots in Figure 11 illustrate the locality (based on RTT) of delivered content from Akamai-owned servers shown at 12 o'clock (labeled as Akamai) as well as from Akamai's guest servers in all three host networks in the snapshot from 10/04/16. The guest servers are grouped by their host ASes and ordered based on their aggregate contribution in delivered

bytes (for Akamai flows) in a clock-wise order. We observe that traffic delivered from Akamai-owned servers exhibits a higher locality – 75% (90%) of the bytes (flows) are delivered from servers that are 4ms (8ms) RTT away. The Akamai traffic from CenturyLink, NTT and OVH is delivered from servers that are at RTT distance of 8, 15 and 20ms, respectively. While these guest servers serve content from further away than the Akamai-owned servers, they are all relatively close to UOnet which suggests that they are not intended to offer higher level of locality for delivered content to UOnet users.

3.6 Implications of Traffic Locality

Improving end user-perceived performance (i.e. decreasing delay and/or increasing throughput) is one of the main motivations for major content providers to bring their front-end servers closer to the edge of the network. In the following, we examine whether such performance improvements are indeed experienced by the end users served by UOnet and to what extent for a given content provider the observed performance is correlated with that content provider's traffic locality.

We already showed in Figure 9 that the measured min RTT values for a majority of content providers (with some exceptions such as OVH, Quantil, Cogent) are consistently low (<20ms) across all flows. The average throughput of each flow can be easily estimated by dividing the total number of delivered bytes by its duration [3]. To get an overall sense of the observed average throughput, Figure 12 shows the summary distributions of the measured throughput across delivered flows by each target content provider. We observe that 90% of the flows for all target content providers (except Level3) experience low throughput (< 0.5MB/s, and in

[3]Note that we may have fragmented flows for this analysis. This means that long flows will be divided into 5min intervals. However, 5min is sufficiently long to estimate average throughput of individual flows.

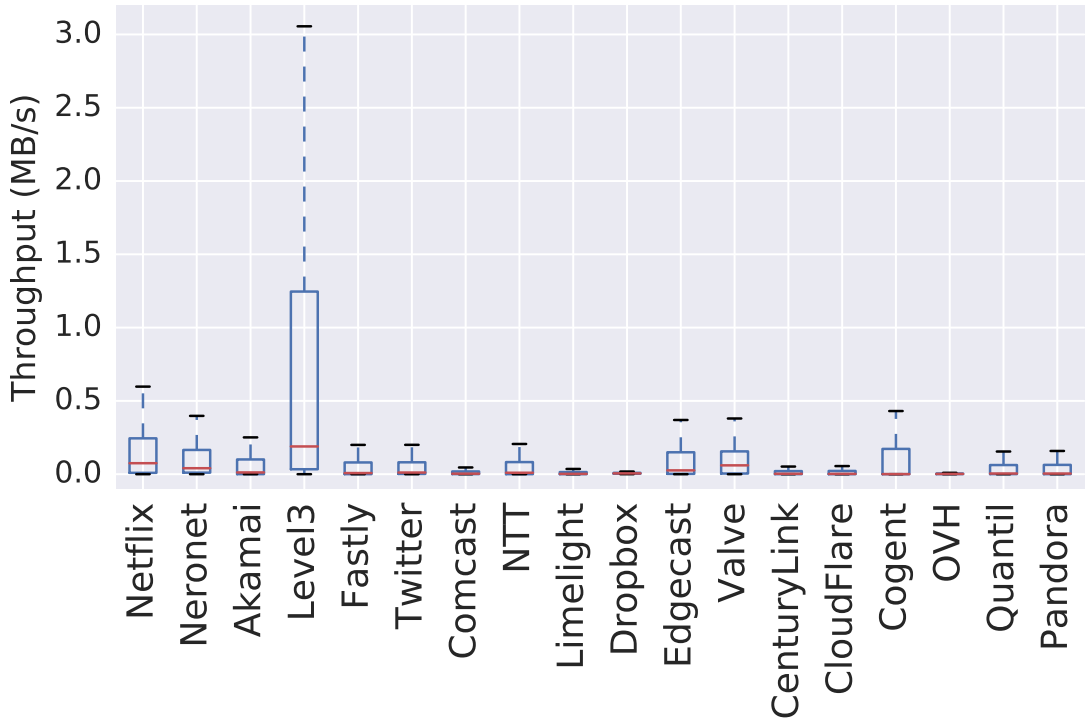


Figure 12. Summary distribution of average throughput for delivered flows from individual target content providers towards UOnet users across all of our snapshots.

most cases even < 0.25MB/s). This raises the question why these very localized flows do not achieve higher throughput.

In general, reliably identifying the main factors that limit the throughput of individual flows is challenging Sundaresan, Feamster, and Teixeira (2015). The cause could be any combination of factors that include

- *Content Bottleneck*: the flow does not have sufficient amount of content to "fill the pipe";
- *Receiver Bottleneck*: the receiver's access link (i.e. client type) or flow control is the limiting factor;
- *Network Bottleneck*: the fair share of network bandwidth is limited due to cross traffic (and resulting loss rate);

– *Server Rate Limit*: a content provider's server may limit its transmission rate implicitly due to its limited capacity or explicitly as a results of the bandwidth requirements of the content (e.g. Netflix videos do not require more than 0.6 MB/s for a Full-HD stream Netflix (2017a)).

Rather than inferring the various factors that affect individual flows, our goal is to identify the primary factor from the above list that limits the maximum achievable throughput by individual content providers. To this end, we only consider 3-4% (or 510-570K) of all flows for each target content provider that their size exceeds 1 MB and refer to them as "elephant" flows.[4] These elephant flows have typically several 100s of packets and are thus able to fully utilize available bandwidth in the absence of other limiting factors (i.e. content bottleneck does not occur). More than 0.5 million elephant flows for individual content providers are delivered to end users in UOnet that have diverse connection types (wireless, residential, wired). Therefor, receiver bottleneck should not be the limiting factor for the maximum achievable throughput by individual content providers. This in turn suggests that either the network or the server are responsible for limiting the achievable throughput.

To estimate the *Maximum Achievable Throughput (MAT)* for each content provider, we group all elephant flows associated with that content provider based on their RTT into 2ms bins and select the 95% throughput value (i.e. median of the top 10%) in the bin as its MAT with its mid-bin RTT value as the corresponding RTT. Since a majority (96%) of these flows are associated with TCP connection and thus are congestion controlled, we can examine the key factors responsible for limiting throughput. Figure 13 shows a scatter plot where each labelled dot

[4]Selecting the 1 MB threshold for flow size strikes a balance between having sufficiently large flows Sodagar (2011) and obtaining a large set of flows for each content provider.

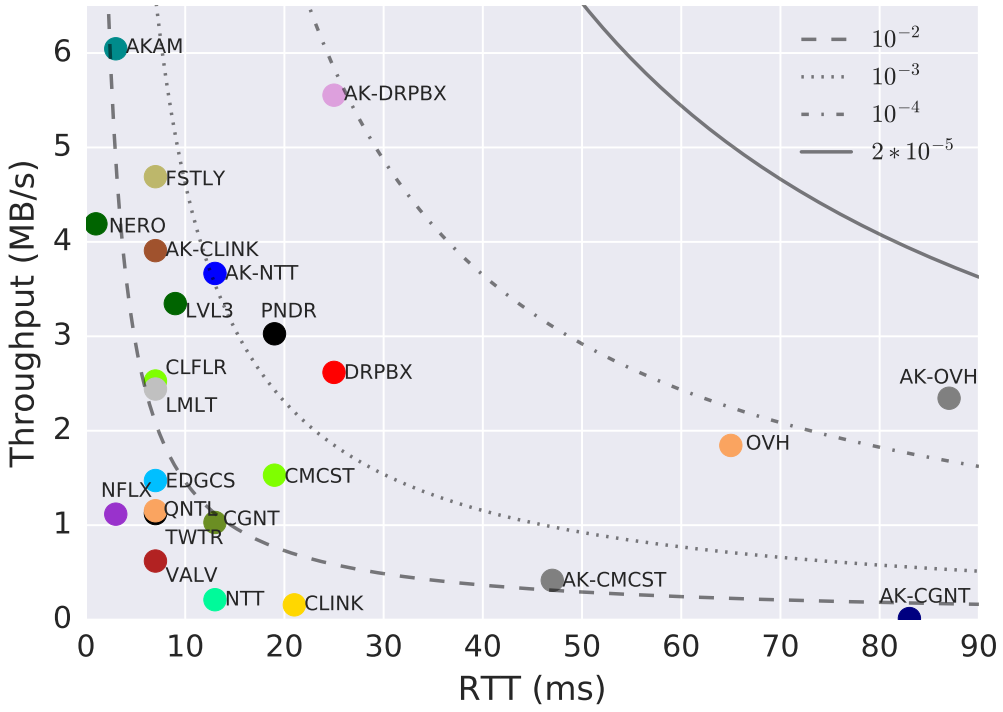


Figure 13. Maximum Achievable Throughput (MAT) vs MinRTT for all content providers. The curves show the change in the estimated TCP throughput as a function of RTT for different loss rates.

represents a target content provider with its y-value denoting its MAT and its x-value denoting the associated RTT. We also group all Akamai flows from its guest servers at each host AS_x, determine their separate MAT and exclude them from AS_x's own flows to avoid double-counting them. For example, Akamai flows that are delivered from OVH are marked as AK-OVH. To properly compare the measured MAT values across different RTTs, we also plot an estimated TCP throughput as a function of RTT for three different loss rates that we obtain by applying the commonly-used equationMathis et al. (1997): $T < \frac{MSS}{RTT} * \frac{1}{\sqrt{L}}$. In this equation, MSS denotes the Maximum Segment Size which we set to 1460; L represents the loss rate. We consider three different loss rate values, namely 10^{-2}, 10^{-3}, 10^{-4}, to cover a wide range of "realistic" values.

Examining Figure 13, we notice that the relative location of each labelled dot with respect to the TCP throughput lines reveals the average “virtual" loss rate across all elephant flows of a content provider if bandwidth bottleneck were the main limiting factor. The figure shows that this virtual loss rate for many content providers is at or above 10^{-3}. However, in practice, average loss rates higher than 10^{-3} over such short RTTs (<20ms) are very unlikely in our setting (e.g., UOnet is well provisioned and most incoming flows traverse the paths with similar or identical tail ends). To test this hypothesis, we directly measure the loss rate between UOnet and the closest servers for each content provider using 170K ping probes per content provider.[5] Figure 14 depicts the average loss rate for each target content provider and shows that the measured average loss rate for all of the target content providers is at least an order of magnitude lower than the virtual loss rate for each content provider. This confirms that all of the measured MAT values must be rate-limited by the server, either explicitly (due to the bandwidth requirements of the content) or implicitly (due to server overload).

Figure 13 also shows that the measured MAT values for Akamai guest servers are often much larger than those for the servers owned by the host AS. For example, the MAT value for AK-CLINK (AS-DRPBX or AK-NTT) is much higher than the MAT for CLINK (DRPBX, or NTT). Furthermore, the measured MAT value for all the flows from Akamai’s guest servers is lower than its counterpart for all flows from Akamai-owned servers.

To summarize, there are two main take-aways from our examination of the performance implications of traffic locality. On the one hand, traffic locality is key to achieving the generally and uniformly very small measured delays for traffic

[5]Note that ping measures loss in both directions of a connection.

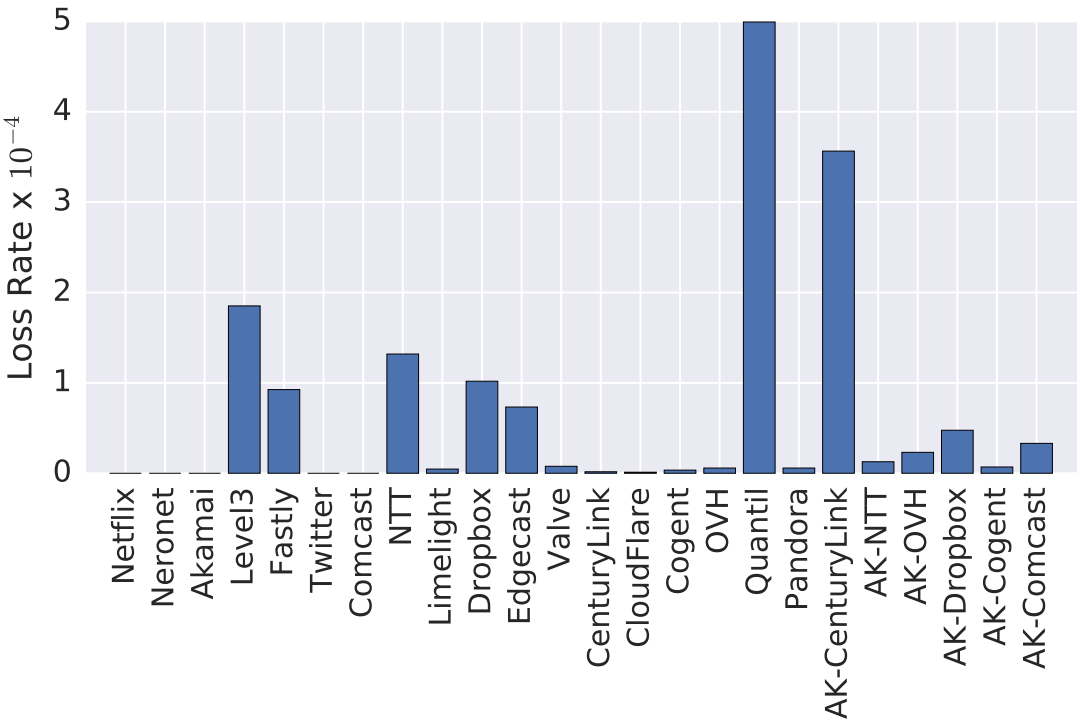


Figure 14. Average loss rate of closest servers per target content provider measured over 24 hours using ping probes with 1 second intervals. For each content provider we choose at most 10 of the closest IP addresses.

delivered to UOnet. On the other hand, our results show that a majority of flows for all target content providers are associated with small files and thus do not reach a high throughput. Furthermore, the throughput for most of the larger flows are not limited by the network but rather by the front-end servers. In other words, high throughput delivery of content at the edge is either not relevant (for small objects) or not required by applications.

3.7 Summary

Our work contributes to the existing literature on content delivery by providing a unique view of different aspects of content delivery as experienced by the end users served by a stub-AS (i.e., a Research & Education network). To this end, we examine the complete flow-level view of traffic delivered to this stub-AS

from all major content providers and characterize this stub-AS' *traffic footprint* (i.e. a detailed assessment of the locality properties of the delivered traffic).

We also study the impact that this traffic footprint has on the performance experienced by its the end users and report on two main takeaways. First, this stub-AS' traffic locality is uniformly high across the main CPs; i.e., the traffic that these CPs deliver to this stub-AS experiences in general only very small delays. Second, the throughput of the delivered traffic remains far below the maximum achievable throughout and is not limited by the network but rather by the front-end servers.

Lastly, to complement the effort described in this chapter, assessing the locality properties of the traffic that constitutes the (long) tail of the distribution in Figure 6 and is typically delivered from source IP addresses that are rarely seen in our data or are responsible for only minuscule portions of the traffic delivered to UOnet looms as an interesting open problem and is part of future work.

CHAPTER IV

CLOUD PEERING ECOSYSTEM

Chapter III presented an overview of CPs and content providers' share in Internet traffic and the degree of locality for their infrastructure. In this chapter, we focus on the topology and connectivity of CPs to the rest of the Internet. We pay special attention to the new form of peering relationships that CPs are forming with edge networks.

The content in this chapter is derived entirely from Yeganeh, Durairajan, Rejaie, and Willinger (2019) as a result of collaboration with co-authors listed in the manuscript. Bahador Yeganeh is the primary author of this work and responsible for conducting all measurements and producing the presented analyses.

4.1 Introduction

In this chapter, we present a third-party, cloud-centric measurement study aimed at discovering and characterizing the unique peerings (along with their types) of Amazon, the largest cloud service provider in the US and worldwide. Each peering typically consists of one or multiple (unique) interconnections between Amazon and a neighboring Autonomous System (AS) that are typically established at different colocation facilities around the globe. Our study only utilizes publicly available information and data (i.e. no Amazon-proprietary data is used) and is therefore also applicable for discovering the peerings of other large cloud providers.[1]

We start by presenting the required background on Amazon's serving infrastructure, including the different types of peerings an enterprise network can establish with Amazon at a colo facility in § 4.2. § 4.3 describes the first round of our data collection; that is, launching cloud-centric traceroute probes from different

[1] As long as the cloud provider does not filter traceroute probes.

regions of Amazon's infrastructure toward all the /24 (IPv4) prefixes to infer a subset of Amazon's peerings. We present our methodology for inferring Amazon's peerings across the captured traceroutes in § 4.4.1. Our second round of data collection consists of using traceroute probes that target the prefixes around the peerings discovered in the first round and are intended to identify all the remaining (IPv4) peerings of Amazon (§ 4.4.2). In § 4.5, we present a number of heuristics to resolve the inherent ambiguity in inferring the specific traceroute segment that is associated with a peering. We further confirm our inferred peerings by assessing the consistency of border interfaces at both the Amazon side and client side of an inferred interconnection.

Pinning (or geo-locating) each end of individual interconnections associated with Amazon's peerings at the metro level forms another contribution of this study (§ 4.6). To this end, we develop a number of methods to identify border interfaces that have a reliable location and which we refer to as anchors. Next, we establish a set of co-presence rules to conservatively propagate the location of anchors to other close-by interfaces. We then identify the main factors that limit our ability to pin all border interfaces at the metro level and present ways to pin most of the interfaces at the regional level. Finally, we evaluate the accuracy and coverage of our pinning technique and characterize the pinned interconnections.

The final contribution of this work is a new method for inferring the client border interface that is associated with that client's VPI with Amazon. In particular, by examining the reachability of a given client border interface from a number of other cloud providers (§ 4.7) and identifying overlapping interfaces between Amazon and those other cloud providers, our method provides a lower bound on the number of Amazon's VPIs. We then assign all inferred Amazon

peerings to different groups based on their key attributes such as being public or private, visible or not visible in BGP, and physical or virtual. We then carefully examine these groups of peerings to infer their purpose and explore hybrid peering scenarios. In particular, we show that one-third of Amazon's inferred peerings are either virtual or not visible in BGP and thus hidden from public measurement. Finally, we characterize the inferred Amazon connectivity graph as a whole.

4.2 Background

Amazon's Ecosystem. The focus of our study of peerings in today's Internet is Amazon, arguably the largest cloud service provider in the US and worldwide. Amazon operates several data centers worldwide. While these data centers' street addresses are not *explicitly* published by Amazon, their geographic locations have been reported elsewhere Burrington (2016); DatacenterMap (2018); Miller (2015); Plaven (2017); WikiLeaks (2018); Williams (2016). Each data center hosts a large number of Amazon servers that, in turn, host user VMs as well as other services (e.g. Lambda). Amazon's data center locations are divided into independent and distinct geographic *regions* to achieve fault tolerance/stability. Specifically, each region has multiple, isolated *availability zones (AZs)* that provide redundancy and offer high availability in case of failures. *AZs* are virtual and their mapping to a specific location within their region is not known Amazon (2018f). As of 2018, Amazon had 18 regions (55 *AZs*) across the world, with five of them (four public + one US government cloud) located in the US. For our study, we were not able to utilize three of these regions. Two of them are located in China, are not offered on Amazon's AWS portal, and require approval requests by Amazon staff. The third region is assigned to the US government and is not offered to the public.

Peering with Amazon at Colo Facilities. Clients can connect to Amazon through a specific set of colo facilities. Amazon is considered a *native* tenant in these facilities, and their locations are publicly announced by Amazon Amazon (2018d). Amazon is also *reachable* through a number of other colo facilities via layer-2 connectivity offered by third-party providers (e.g. Megaport).[2]

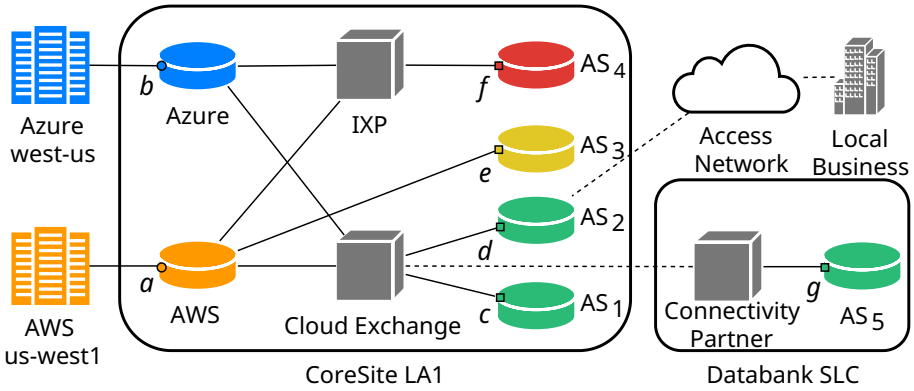


Figure 15. Overview of Amazon's peering fabric. Native routers of Amazon & Microsoft (orange & blue) establishing private interconnections (AS_3 - yellow router), public peering through IXP switch (AS_4 - red router), and virtual private interconnections through cloud exchange switch (AS_1, AS_2, and AS_5 - green routers) with other networks. Remote peering (AS_5) as well as connectivity to non-ASN businesses through layer-2 tunnels (dashed lines) happens through connectivity partners.

Figure 15 depicts an example of different types of peerings offered by cloud providers at two colo facilities. Both Amazon (AWS) and Microsoft (Azure) are native (i.e. house their border routers) in the CoreSite LA1 colo facility and are both present at that facility's IXP and cloud exchange. (Open) cloud exchanges are switching fabrics specifically designed to facilitate interconnections among network providers, cloud providers, and enterprises in ways that provide the scalability and elasticity essential for cloud-based services and applications (e.g. see CoreSite (2018); Equinix (2017)). Major colo facility providers (e.g. Equinix

[2]These entities are called "AWS Direct Connect Partners" at a particular facility and are listed online along with their points of presence Amazon (2018c).

and CoreSite) also offer a new interconnection service option called "virtual private interconnection (VPI)." VPIs enable local enterprises (that may or may not own an ASN) to connect to multiple cloud providers that are present at the cloud exchange switching fabric by means of purchasing a single port on that switch. In addition, VPIs provide their customers access to a programmable, real-time cloud interconnection management portal. Through this portal, the operators of these new switching fabrics make it possible for individual enterprises to establish their VPIs in a highly-flexible, on-demand, and near real-time manner. This portal also enables enterprises to monitor in real-time the performance of their cloud-related traffic that traverses these VPIs.

While cloud exchanges rely on switching fabrics that are similar to those used by IXPs, there are two important differences. For one, cloud exchanges enable each customer to establish virtualized peerings with multiple cloud providers through a single port. Moreover, they provide exclusive client connectivity to cloud providers without requiring a client to use its pre-allocated IP addresses. Operationally, a cloud customer establishes VPIs using either public or private IP addresses depending on the set of cloud services that this customer is trying to reach through these interconnections. On the one hand, VPIs relying on private addresses are limited to the customer's virtual private cloud (VPC) through VLAN isolation. On the other hand, VPIs with public addresses can reach compute resources in addition to other AWS offerings such as S3 and DynamoDB Amazon (2018b). Given the isolation of network paths for VPIs with private addresses, any peerings associated with these VPIs are not visible to the probes from VMs owned by other Amazon customers. This makes it, in practice, impossible to discover established VPIs that rely on private addresses. In Figure 15, the different colors

of the client routers indicate the type of their peerings; e.g. public peering through the IXP (for AS_4), direct physical interconnection (also called "cross-connect") (for AS_3), private virtual peerings that are either local (for AS_1 and AS_2) or remote (for AS_5). Here, a local virtual private peering (e.g. AS_2) could be associated with an enterprise that is brought to the cloud exchange by its access network (e.g. Comcast) using layer-2 technology; based on traceroute measurements, such a peering would appear to be between Amazon and the access network. In contrast, a remote private virtual peering could be established by an enterprise (e.g. AS_5) that is present at a colo facility (e.g. Databank in Salt Lake City in Figure 15) where Amazon is not native but that houses an "AWS Direct Connect Partner" (e.g. Megaport) which in turn provides layer-2 connectivity to AWS.

4.3 Data Collection & Processing

To infer all peerings between Amazon and the rest of the Internet, we perform traceroute campaigns from Amazon's 15 available global regions to a `.1` in each `/24` prefix of the IPv4 address space.[3] To this end, we create a *t2-micro* instance VM within each of the 15 regions and break down the IPv4 address space into `/24` prefixes. While we exclude broadcast and multicast prefixes, we deliberately consider addresses that are associated with private and shared address spaces since these addresses can be used internally in Amazon's own network. This process resulted in 15.6M target IPv4 addresses.

To probe these target IPs from our VMs, we use the Scamper tool Luckie (2010) with UDP probes as they provide the highest visibility (i.e. response rate). Individual probes are terminated upon encountering five consecutive unresponsive hops in order to limit the overall measurement time while reaching

[3]We observed a negligible difference in the visibility of interconnections across probes from different *AZs* in each region. Therefore, we only consider a single *AZ* from each region.

Amazon's border routers. We empirically set our probing rate to 300pps to prevent blacklisting or rate control of our probe packets by Amazon. With this probing rate, our traceroute campaign took nearly 16 days to complete (from 08/03/2018 to 08/19/2018). Each collected traceroute is associated with a status flag indicating how the probe was terminated. We observed that the total number of completed traceroutes across different regions is fairly consistent but rather small (mean 7.7% and std $5 * 10^{-4}$) which suggests a limited yield. However, since our main objective is to identify Amazon interconnections and *not* to maximize traceroute yield, we consider any traceroute that leaves Amazon's network (i.e. reaches an IP outside of Amazon's network) as a candidate for revealing the presence of an interconnection, and the percentage of these traceroutes is about 77%.

Annotating Traceroute Data. To identify any Amazon interconnection traversed by our traceroutes, we annotate every IP hop with the following information: *(i)* its corresponding ASN, *(ii)* its organization (ORG), and *(iii)* whether it belongs to an IXP prefix. To map each IP address to its ASN, we rely on BGP snapshots from RouteViews and RIPE RIS (taken at the same time as our traceroute campaign). For ORG, we rely on CAIDA's AS-to-ORG dataset Huffaker, Keys, Fomenkov, and Claffy (2018) and map the inferred ASN of each hop from the previous step to its unique ORG identifier. ORG information allows us to correctly identify the border interface of a customer in cases where traceroute traverses through hops in multiple Amazon ASes prior to reaching a customer network[4]. Finally, to determine if an IP hop is part of an IX prefix, we rely on PeeringDB PeeringDB (2017), Packet Clearing House (PCH) Packet Clearing House

[4]We observed AS7224, AS16509, AS19047, AS14618, AS38895, AS39111, AS8987, and AS9059 for Amazon.

(2017), and CAIDA's IXP dataset CAIDA (2018) to obtain prefixes assigned to IXPs.

In our traceroutes, we observe IP hops that do not map to any ASN. These IPs can be divided into two groups. The first group consists of the IPs that belong to either a private or a shared address space (20.3%); we set the ASN of these IPs to 0. The second group consists of all the IPs that belong to the public address space but were not announced by any AS during our traceroute campaign (7%); for these IPs, we infer the AS owner by relying on WHOIS-provided information (i.e. name or ASN of the entity/company assigned by an RIR).

4.4 Inferring Interconnections

In this section, we describe our basic inference strategy for identifying an Amazon-related interconnection segment across a given traceroute probe (§ 4.4.1) and discuss the potential ambiguity in the output of this strategy. We then discuss the extra steps we take to leverage these identified segments in an effort to efficiently expand the number of discovered Amazon-related interconnections (§ 4.4.2).

4.4.1 Basic Inference Strategy. Given the ASN-annotated traceroute data, we start from the source and sequentially examine each hop until we detect a hop that belongs to an organization *other* than Amazon (i.e. its ORG number is neither 0 nor 7224, which is Amazon). We refer to this hop as *customer border hop* and to its IP as a *Customer Border Interface (CBI)*. The presence of a *CBI* indicates that the traceroute has exited Amazon's network; that is, the traceroute hop right before a *CBI* is the *Amazon Border Interface (ABI)*, and the corresponding traceroute probe thus must have traversed an Amazon-related *interconnection segment*. For the remainder of our analysis, we only consider these

initial portions of traceroutes between a source and an encountered *CBI*.[5] Next, for each *CBI*, we check to confirm that the AS owners of all the downstream hops in each traceroute does not include any ASN owned by Amazon (i.e. a sanity check that the traceroute does not re-enter Amazon); all of our traceroutes meet this condition. Finally, because of their unreliable nature, we exclude all traceroutes that contain either an (IP-level) loop, unresponsive hop(s) prior to Amazon's border, a *CBI* as the destination of a traceroute Baker (1995), or duplicate hops before Amazon's border. The first two rows of Table 3 summarize the number of *ABIs* and *CBIs* that we identified in our traceroute data, along with the fraction of interfaces in each group for which we have BGP, Whois, and IXP-association information. As highlighted in § 2.3.2, in certain cases, our basic strategy may not identify the correct Amazon-related interconnection segment on a given traceroute. Given that our traceroutes are always launched from Amazon to a client's network, when Amazon provides addresses for the physical interconnection, our strategy incorrectly identifies the next downstream segment as an interconnection Amazon (2018b).

In summary, the described method always reveals the presence of an Amazon-related interconnection segment in a traceroute. The actual Amazon-specific interconnection segment is either the one between the identified ABI and CBI or the immediately preceding segment. Because of this ambiguity in accurately inferring the Amazon-specific interconnection segments, we refer to them as candidate *interconnection segments. In § 4.5, we present techniques for a more precise determination of these inferred candidate interconnection segments.*

[5]In fact, we only need the *CBI* and the prior two *ABIs*.

Table 3. Number of unique *ABIs* and *CBIs* along with their fraction with various meta data, prior (rows 2-3) and after (rows 4-5) /24 expansion probing.

	All	BGP%	Whois%	IXP%
ABI	3.68k	38.4%	61.6%	-
CBI	21.73k	54.74%	24.8%	20.46%
eABI	3.78k	38.85%	61.15%	-
eCBI	24.75k	79.82%	22.32%	17.86%

4.4.2 Second Round of Probing to Expand Coverage. We perform our traceroute probes from each Amazon's region in two rounds. First, as described in § 4.4.1, we target `.1` in each `/24` prefix of the IPv4 address space (§ 4.3) and identify the pool of candidate interconnection segments. However, it is unlikely that our traceroute probes in this first round traverse through all the Amazon interconnections. Therefore, to increase the number of discovered interconnections, in a second round, we launch traceroutes from each region towards all other IP addresses in the `/24` prefixes that are associated with each *CBI* that we discovered in the first round. Our reasoning for this "expansion probing" is that the IPs in these prefixes have a better chance to be allocated to *CBIs* than the IPs in other prefixes. Similar to round one, we annotate the resulting traceroutes and identify their interconnection segments (and the corresponding *ABIs* and *CBIs*). The bottom two rows in Table 3 show the total number of identified *ABIs* and *CBIs* after processing the collected expansion probes. In particular, while the first column of Table 3 shows a significant increase in the number of discovered *CBIs* (from 21.73k to 24.99k) and even some increase in the number of peering

ASNs (from 3.52k to 3.55k) as a result of the expansion probing, the number of *ABIs* remains relatively constant.

4.5 Verifying Interconnections

To address the potential ambiguity in identifying the correct Amazon-specific segment of each inferred interconnection (§ 4.4.1), we first check these interconnections against three different heuristics (§ 4.5.1) and then rely on the router-level connectivity among border routers (§ 4.5.2) to verify (and possibly correct) the inferred *ABIs* and *CBIs*.

4.5.1 Checking Against Heuristics. We develop a few heuristics to check the aforementioned ambiguity of our approach with respect to inferring the correct interconnection segment. Since the actual interconnection segment could be the segment prior to the identified candidate segment (i.e. we might have to shift the interconnection to the previous segment), our heuristics basically check for specific pieces of evidence to decide whether an inferred *ABI* is correct or should be changed to its corresponding *CBI*. Once an *ABI* is confirmed, all of its corresponding *CBIs* are also confirmed. The heuristics are described below and are ordered (high to low) based on our level of confidence in their outcome.

IXP-Client. An IP address that is part of an IXP prefix always belongs to a specific IXP member. Therefore, if the IP address for a *CBI* in a candidate interconnection segment is part of an IXP prefix, then that *CBI* and its corresponding *ABI* are correctly identified Nomikos and Dimitropoulos (2016).

Hybrid IPs. We observe *ABI* interfaces with hybrid connectivity. For example, in Figure 16, interface a represents such an interface with hybrid connectivity; it appears prior to the client interface b in one traceroute and prior to the Amazon interface c in another traceroute. Even if we are uncertain about the owner of

an interface c (i.e. it may belong to the same or different Amazon client), we can reliably conclude that interface a has hybrid connectivity and must be an *ABI*.

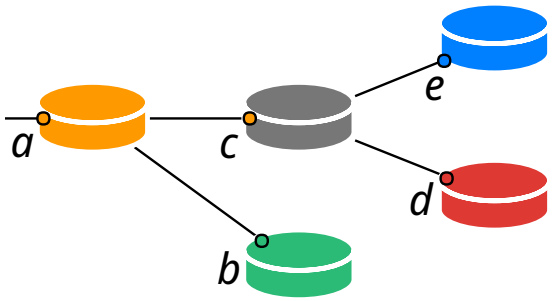


Figure 16. Illustration of a hybrid interface (a) that has both Amazon and client-owned interfaces as next hop.

Interface Reachability. Our empirical examination of traceroutes revealed that while *ABIs* are generally reachable from their corresponding clients, for security reasons, they are often not visible/reachable from the public Internet (e.g. a campus or residential networks). However, depending on the client configuration, *CBIs* may or may not be publicly reachable. Based on this empirical observation, we apply a heuristic that probes all candidate *ABIs* and *CBIs* from a vantage point in the public Internet (i.e. a node at the University of Oregon). Reachability (or unreachability) of a candidate *CBI* (or *ABIs*) from the public Internet offers independent evidence in support of our inference.

Table 4 summarizes the fraction of identified *ABIs* (and thus their corresponding *CBIs*) that are confirmed by our individual (first row) and combined (second row) heuristics, respectively. We observe that our heuristics collectively confirmed 87.8% of all the inferred *ABIs* and thus 96.96% of the *CBIs*. The remaining 0.37k (or 9.81%) *ABIs* that do not match with any heuristic are interconnected with one (or multiple) *CBIs* that belong to a single organization. The resulting low rate of error in detecting the correct interconnection segments implies high confidence in the correctness of our inferred Amazon peerings.

Table 4. Number of candidate *ABIs* (and corresponding *CBIs*) that are confirmed by individual (first row) and cumulative (second row) heuristics.

	IXP	**Hybrid**	**Reachable**
Individual	0.83k (13.66k)	2.05k (14.44k)	2.8k (15.14k)
Cumulative	0.83k (13.66k)	2.26k (15.14k)	3.31k (24.23k)

4.5.2 Verifying Against Alias Sets. To further improve our ability to eliminate possible ambiguities in inferring the correct interconnection segments, we infer the router-level topology associated with all the candidate interconnections segments and determine the AS owner of individual routers. We consider any inferred interconnection segment to be correct if its *ABI* is on an Amazon router and its *CBI* is on a client router. In turn, for any incorrect segment, we first adjust the ownership of its corresponding *ABI* and *CBI* so as to be consistent with the determined router ownership and then identify the correct interconnection segment.

To this end, we utilize MIDAR Bender et al. (2008) to perform alias resolution from VMs in all the regions where all the candidate *ABIs* and *CBIs* were observed. Each instance of this alias resolution effort outputs a set of (two or more) interfaces that reside on a single router. Given the potentially limited visibility of routers from different regions, we combine the alias sets from different regions that have any overlapping interfaces. Overall, we identify 2.64k alias sets containing 8.68k (2.31k *ABI* plus 6.37k *CBI*) interfaces and their sizes have a skewed distribution.

The direction of our traceroute probes (from Amazon towards client networks) and the fact that each router typically responds with the incoming interface suggest that the observed interfaces of individual Amazon (or client)

border routers in our traceroute (i.e. IPs in each alias set) should typically belong to the same AS. This implies that there should be a majority AS owner among interfaces in an alias set. To identify the AS owner of each router, we simply examine the AS owner of individual IPs in the corresponding alias set. The AS that owns a clear majority of interfaces in an alias set is considered as the owner of the corresponding router and all the interfaces in the alias set.[6] We observe that for more than 94% (92%) of all alias sets, there is a single AS that owns >50% (100%) of all of an alias set's interfaces. The remaining 6% of alias sets comprises 343 interfaces with a median set size of 2. We consider the majority AS owner of each alias set as the AS owner of (all interfaces for) that router. Using this information, we check all of the inferred *ABIs* and *CBIs* to ensure that they are on a router owned by Amazon and the corresponding client, respectively. Otherwise, we change their labels. This consistency check results in changing the status of only 45 interfaces (i.e. 18, 2, and 25 change from $ABI \rightarrow CBI$, $CBI \rightarrow ABI$, and $CBI \rightarrow CBI$[7], respectively). *These changes ultimately result in 3.77k ABIs and 24.76k CBIs associated with 3.55k unique ASes.*

4.6 Pinning Interfaces

In this section, we first explore techniques to pin (i.e. geo-locate) each end of the inferred Amazon peerings (i.e. all *ABIs* and *CBIs*) to a specific colo facility, metro area, or a region and then evaluate our pinning methodology.

4.6.1 Methodology for Pinning. Our method for pinning individual interfaces to specific locations involves two basic steps. In a first step, we identify

[6]We also examined router ownership at the organization level by considering all ASNs that belong to a single organization. This strategy allows us to group all Amazon/client interfaces regardless of their ASN to accurately detect the AS owner. However, since we observed one ASN per ORG in 99% of the identified alias sets, we present here only the owner AS of each router.

[7]This simply implies that the *CBI* interface belongs to another client.

a set of border interfaces with known locations that we call *anchors*. Then, in a second step, we establish two *co-presence* rules to iteratively infer the location of individual unpinned interfaces based on the location of co-located anchors or other already pinned interfaces. That is, in each iteration, we propagate the location of pinned interfaces to their co-located unpinned neighbors.

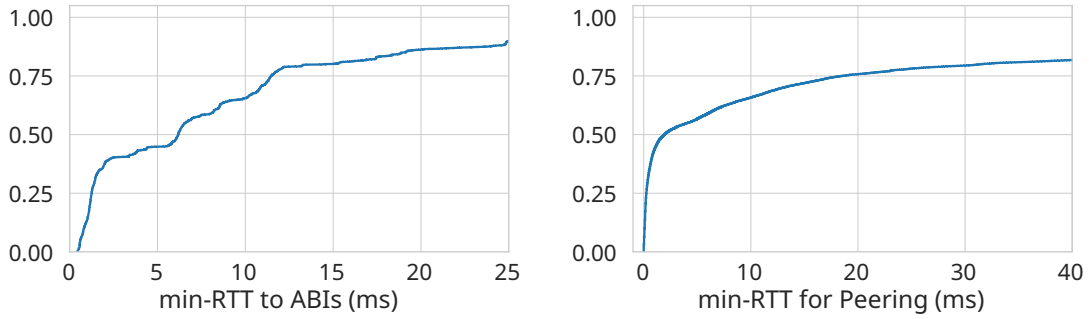
Identifying Anchors. For *ABIs* or *CBIs* to serve as anchors for pinning other interfaces, we leverage the following four sources of information and consider them as reliable indicators of interface-specific locations.

DNS Information (CBIs): A *CBI*[8] with specific location information embedded in its DNS name can be pinned to the corresponding colo or metro area. For example, a DNS name such as `ae-4.amazon.atlnga05.us.bb.gin.ntt.net` indicates that the *CBI* associated with NTT interconnects with Amazon in Atlanta, GA (*atlnga*). We use DNS parsing tools such as DRoP Huffaker, Fomenkov, and claffy (2014) along with a collection of hand-crafted rules to extract the location information (using 3-letter airport codes and full city names) from the DNS names of identified *CBIs*. In the absence of any ground truth, we check the inferred geolocation against the footprint of the corresponding AS from its PeeringDB listings or information on its webpage. Furthermore, we perform an RTT-constraint check using the measured RTTs from different Amazon regions to ensure that the inferred geolocation is feasible. This check, similar to DRoP Huffaker, Fomenkov, and claffy (2014), conservatively excludes 0.87k *CBIs* for which their inferred locations do not satisfy this RTT constraint.

IXP Association (CBIs): CBIs that are part of an IXP prefix can be pinned to the colo(s) in a metro area where the IXP is present. In total we have identified

[8]None of the *ABIs* had a reverse domain name associated with them.

671 IXPs within 471 (117) unique cities (countries) but exclude 10 IXPs (and their corresponding 366 *CBIs*) that are present in multiple metro areas as they cannot be pinned to a specific colo or metro area. Furthermore, we exclude all interfaces belonging to members that peer remotely. To determine those members, we first identified minIXRegion, the closest Amazon region to each IXP. We did this by measuring minIXRTT, the minimum RTT between the various regions and all interfaces that are part of the IXP and selecting minIXRegion as the Amazon region where minIXRTT is attained. Then we measure the minimum RTT between all interfaces and minIXRegion and label an interface as "local" if its RTT value is no more than 2ms higher than minIXRTT. We note that for about 80% of IXPs, the measured minIXRTT is less than 1.5ms (i.e. most IXPs are in very close proximity to at least one AWS region). This effort results in labeling about 2k out of the encountered 3.5k IXP interfaces in our measurements as "local." Conversely, there are some 1.5k interfaces belonging to members that peer remotely.



(a) (b)

Figure 17. (a) Distribution of min-RTT for *ABIs* from the closest Amazon region, and (b) Distribution of min-RTT difference between *ABI* and *CBI* for individual peering links.

Single Colo/Metro Footprint (CBIs): CBIs of an AS that are present only at a single colo or at multiple colos in a given metro area can be pinned to that metro area. To identify those ASes that are only present in a single colo or a single metro

area, we collect the list of all tenant ASes for 2.6k colo facilities from PeeringDB Lodhi, Larson, Dhamdhere, Dovrolis, et al. (2014) as well as the list of all IXP participants from PeeringDB and PCH.

Native Amazon Colos (ABIs): Intuitively, *ABIs* that are located at colo facilities where Amazon is native (i.e. facilities that house Amazon's main border routers) must exhibit the shortest RTT from the VM in the corresponding region. To examine this intuition, we use two data sources for RTT measurements: *(i)* RTT values obtained through active probing [9] of *CBIs* and *ABIs*; and *(ii)* RTT values collected as part of the traceroute campaign. Figure 17a shows the distribution of the minimum RTT between VMs in different regions of Amazon and individual *ABIs*. We observe a clear knee at 2ms where around 40% of all the *ABIs* exhibit shorter RTT from a single VM. Given that all Amazon peerings have to be established through colo facilities where Amazon is native, we pin all these *ABIs* to the native colo closest to the corresponding VM. In some metro areas where Amazon has more than one native colo, we conservatively pinned the *ABIs* to the corresponding metro area rather than to a specific native colo.

Consistency Checking of Anchors. We perform two sets of consistency checks on the identified anchors. First, we check whether the inferred locations are consistent for those interfaces (i.e. 1.1k in total) that satisfy more than one of the four indicators we used to classify them as anchors. Second, we check for consistency across the inferred geolocation of different interfaces in any given alias set. These checks flagged a total of 66 (48 and 18) interfaces that had inconsistent geolocations and that we therefore excluded from our anchor list. These checks also highlight the conservative nature of our approach. In particular, by removing any

[9]This probing was done for a full day and used exclusively ICMP echo reply messages that can only be generated by intermediate hops and not by the target itself.

anchors with inconsistent locations, we avoid the propagation of unreliable location information in our subsequent iterative pinning procedure (see below). The middle part of Table 5 presents the exclusive and cumulative numbers of *CBI* and *ABI* anchors (excluding the flagged ones) that resulted from leveraging the four utilized source of information.

Inferring Co-located Interfaces. We use two co-presence rules to infer whether two interfaces are co-located in the same facility or same metro area. *(i) Rule 1 (Alias sets)*: This rule states that all interfaces in an alias set must be co-located in the same facility. Therefore, if an alias set contains one (or more) anchor(s), all interfaces in that set can be pinned to the location of that (those) anchor(s). *(ii) Rule 2 (Interconnections in a Single Metro Area)*: An Amazon peering is established between an Amazon border router and a client border router, and these routers are either in the same or in different colo/metro areas. Therefore, a small RTT between the two ends of an interconnection segment is an indication of their co-presence in at least the same metro area. The key issue is to determine a proper threshold for RTT delay to identify these co-located pairs. To this end, Figure 17b shows the distribution of the min-RTT differences between the two ends of all the inferred Amazon interconnection segments. While the min-RTT difference varies widely across all interconnection segments, the distribution exhibits a pronounced knee at 2ms, with approximately half of the inferred interconnection segments having min-RTT values less than this threshold. We use this threshold to separate interconnection segments that reside within a metro area (i.e. both ends are in the metro area) from those that extend beyond the metro area. Therefore, if one end of such a “short" interconnection segment is pinned, its other end can be pinned to the same metro area.

Table 5. The exclusive and cumulative number of anchor interfaces by each type of evidence and pinned interfaces by our co-presence rules.

	Anchor Interface				Pinned Interface	
	DNS	**IXP**	**Metro**	**Native**	**Alias**	**min-RTT**
Exc.	5.31k	2.0k	1.66k	1.42k	0.65k	5.38k
Cum.	5.31k	6.73k	7.22k	8.64k	9.21k	14.37k

Iterative Pinning. Given a set of initial anchors at known locations as input, we identify and pin the following two groups of interfaces in an iterative fashion: *(i)* all unpinned alias sets that contain one (or more) anchor(s), and *(ii)* the unpinned end of all the short interconnection segments that have only one end pinned. For both steps, we extend our pinning knowledge to other interfaces *only if all anchors unanimously agree with the geolocation of the unpinned interface* [10]. This iterative process ends when there is no more interface that meets our co-presence rules. Our pinning process requires only four rounds to complete. The right-hand side of Table 5 summarizes the exclusive and cumulative number of interfaces pinned by each co-presence rule. Including all the anchors, *we are able to pin 45.05% (75.87%) of all the inferred CBIs (ABIs), and 50.21% of all border interfaces associated with Amazon's peerings.*

Pinning at a Coarser Resolution. To better understand the reasons for being able to map only about half of all inferred interfaces associated with Amazon at the metro level, we next explore whether the remaining (14.21k) unpinned interfaces can be associated with a specific Amazon region based on their relative RTT distance. To this end, we examine the ratio of the two smallest min-RTT values for individual unpinned interfaces from each of the 15 Amazon regions. 1.11k of these

[10]We observed such a conflict in the propagation of pinning information only for 179 (1.2%) interfaces

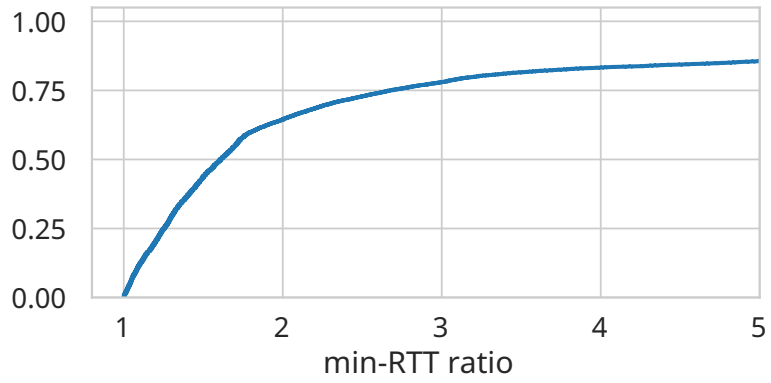


Figure 18. Distribution of the ratio of two lowest min-RTT from different Amazon regions to individual unpinned border interfaces.

interfaces are only visible from a single region and therefore the aforementioned ratio is not defined for these interfaces. We associate these interfaces to the only region from which they are visible. Figure 18 depicts the CDF of the ratio for the remaining (13.1k) unpinned interfaces that are reachable from at least two regions and shows that for 57% of these interfaces the ratio of two lowest min-RTT is larger than 1.5, i.e. the interface's RTT is 50% larger for one region. We map these interfaces to the region with the lowest delay. The relatively balanced min-RTT values for the remaining 43% of interfaces is mainly caused by the limited geographic separation of some regions. For example, the relatively short distance between *Virginia* and *Canada*, or between neighboring European countries makes it difficult to reliably associate some of the interfaces that are located between them using min-RTT values. *This coarser pinning strategy can map 8.67k (30.37%) of the remaining interfaces (0.62k ABIs and 8.05k CBIs) to a specific region which improves the overall coverage of the pinning process to a total of 80.58%.* However, because of the coarser nature of pinning, we do not consider these 30.37% of interfaces for the rest of our analysis and only focus on those 50.21% that we pinned at the metro (or finer) level.

4.6.2 Evaluation of Pinning.

Accuracy. Given the lack of ground truth information for the exact location of Amazon's peering interfaces, we perform cross-validation on the set of identified anchors to enhance the confidence in our pinning results. Specifically, we perform a 10-fold stratified cross-validation with a 70-30 split for train-test samples. We employ stratified sampling Diamantidis, Karlis, and Giakoumakis (2000) to maintain the distribution of anchors within each metro area and avoid cases where test samples are selected from metro areas with fewer anchors. We run our pinning process over the training set and measure both the number of pinned interfaces that match the test set (recall) and the number of pinned interfaces which agree geolocation-wise with the test set (precision). The results across all rounds are very consistent, with a mean value of 99.34% (57.21%) for precision (recall) and a standard deviation of $1.6 * 10^{-3}$ ($5.5 * 10^{-3}$). The relatively low recall can be attributed to the lack of known anchors in certain metro areas that prevented pinning information from propagating. The high precision attests to the conservative nature of our propagation technique (i.e. inconsistent anchors are removed and interfaces are only pinned when reliable (location) information is available) and highlights the low false positive rate of our pinning approach.

Geographic Coverage. We examine the coverage of our pinning results by comparing the cities where Amazon is known to be present against the metros where we have pinned border interfaces. Combining the reported list of served cities by Amazon Amazon (2018d) and the list of PeeringDB-provided cities PeeringDB (2017) where Amazon establishes public or private peerings shows that Amazon is present in 74 metro areas. Our pinning strategy has geo-located Amazon-related border interfaces to 305 different metro areas across the world that cover all but

Table 6. Number (and percentage) of Amazon's VPIs. These are *CBIs* that are also observed by probes originated from Microsoft, Google, IBM, and Oracle's cloud networks.

	Microsoft (%)	**Google (%)**	**IBM (%)**	**Oracle (%)**
Pairwise	4.69k (18.93)	0.79k (3.17)	0.23k (0.94)	0 (0)
Cumulative	4.69k (18.93)	4.93k (19.91)	5.01k (20.23)	5.01k (20.23)

three metro areas from Amazon's list, namely Bangalore (India), Zhongwei (China), and Cape Town (South Africa). While it is possible for some of our discovered, but unpinned *CBIs* to be located in these metros, we lack anchors in these three metros to reliably pin any interface to these locations. Finally, that our pinning strategy results in a significantly larger number of observed metros than the 74 metro areas reported by Amazon should not come as a surprise in view of the many inferred remote peerings where we have sufficient evidence to reliably pin the corresponding *CBIs*.

4.7 Amazon's Peering Fabric

In this section, we first present a method to detect whether an inferred Amazon-related interconnection is virtual (§ 4.7.1). Then we utilize various attributes of Amazon's inferred peerings to group them based on their type (§ 4.7.2) and reason about the differences in peerings across the identified groups (§ 4.7.3). Finally, we characterize the entire inferred Amazon connectivity graph (§ 4.7.4).

4.7.1 Detecting Virtual Interconnections. To identify private peerings that rely on virtual interconnections, we recall that a VPI is associated with a single (*CBI*) port that is utilized by a client to exchange traffic with one or more cloud providers (or other networks) over a layer-2 switching fabric. Therefore,

a *CBI* that is common to two or more cloud providers must be associated with a VPI. Motivated by this observation, our method for detecting VPIs consists of the following three steps. First, we create a pool of target IP addresses that is composed of all identified non-IXP *CBIs* for Amazon, each of their *+1* next IP address, and all the destination IPs of those traceroutes that led to the discovery of individual unique *CBIs*. Second, we probe each of these target IPs from a number of major cloud providers other than Amazon and infer all the *ABIs* and *CBIs* along with the probes that were launched from these other cloud providers (using the methodology described in § 4.4). Finally, we identify any overlapping *CBIs* that were visible from two (or more) cloud providers and consider the corresponding interconnection to be a VPI. Note that this method yields a lower bound for the number of Amazon-related VPIs as it can only identify VPIs whose *CBIs* are visible from the considered cloud service providers. Any VPI that is not used for exchanging traffic with multiple cloud provider is not identified by this method. Furthermore, we are only capable of identifying VPIs which utilize public IP addresses for their *CBIs* Amazon (2018b). VPIs utilizing private addresses are confined to the virtual private cloud (VPC) of the customer and are not visible from anywhere within or outside of Amazon's network.

Applying this method, we probed nearly 327k IPs in our pool of target IP addresses from VMs in all regions of each one of the following four large cloud providers: Microsoft, Google, IBM, and Oracle. The results are shown in Table 6 where the first row shows the number of pairwise common *CBIs* between Amazon and other cloud providers. The second row shows the cumulative number of overlapping *CBIs*. From this table, we observe that roughly 20% of Amazon's *CBIs* are related to VPIs as they are visible from at least one other of the four considered

cloud provider. While roughly 19% of VPIs are common between Amazon and Microsoft, there is no overlap in VPIs between Amazon and Oracle. Only 0.1% of Amazon's *CBIs* are common with Microsoft, Google and IBM.

Note that our method incorrectly identifies a VPI if a customer's border router is directly connected to Amazon but responds to our probe with a default or 3rd party interface. However, either of these two scenarios is very unlikely. For one, recall (§ 4.4) that we use UDP probes and do not consider a target interface as a *CBI* to avoid a response by the default interface Baker (1995). Furthermore, our method selects *+1* IP addresses as traceroute targets (i.e. during the expansion probing) to increase the likelihood that the corresponding traceroutes cross the same *CBI* without directly probing the *CBI* itself. Also, the presence of a customer border router that responds with a third party interface implies that the customer relies on the third party for reaching Amazon while directly receiving downstream traffic from Amazon. However, such a setting is very unlikely for Amazon customers.

4.7.2 Grouping Amazon's Peerings. To study Amazon's inferred peering fabric, we first group all the inferred peerings/interconnections based on the following three key attributes: *(i)* whether the type of peering relationship is public or private, *(ii)* whether the corresponding AS link is present in public BGP feeds, and *(iii)* in the case of private peerings, whether the corresponding interconnection is physical or virtual (VPI). A peering is considered to be public (bi-lateral or multi-lateral) if its *CBI* belongs to an IXP prefix. We also check whether the corresponding AS relationship is present in the public BGP data by utilizing CAIDA's AS Relationships dataset CAIDA (2018) corresponding to the dates of our data collection. Although this dataset is widely used for AS

Table 7. Breakdown of all Amazon peerings based on their key attributes.

Group	**ASes(%)**	**CBIs(%)**	**ABIs(%)**
Pb-nB	2.52k (71)	3.93k (16)	0.79k (21)
Pb-B	0.20k (5)	0.56k (2)	0.56k (15)
Pb	*2.69k (76)*	*4.46k (18)*	*0.83k (22)*
Pr-nB-V	0.24k (7)	2.99k (12)	0.54k (14)
Pr-nB-nV	1.1k (31)	10.24k (41)	2.59k (69)
Pr-nB	*1.18k (33)*	*13.24k (53)*	*2.68k (71)*
Pr-B-nV	0.11k (3)	5.67k (23)	2.07k (55)
Pr-B-V	0.06k (2)	2.09k (8)	0.33k (9)
Pr-B	*0.12k (3)*	*7.76k (31)*	*2.11k (56)*

relationship information, its coverage is known to be limited by the number and placement of BGP feed collectors (e.g., see Luckie, Huffaker, Dhamdhere, Giotsas, et al. (2013) and references therein).

Table 7 gives the breakdown of all of Amazon's inferred peerings into six groups based on the aforementioned three attributes. We use the labels Pr/Pb to denote private/public peerings, B/nB for being visible/not visible in public BGP feeds, and V/nV for virtual/non-virtual peerings (applies only in the case of private interconnections). For example, Pr-nB-nV refers to the number of Amazon's (unique) inferred private peerings that are not seen in public BGP feeds and are not virtual (e.g. cross connections). Each row in Table 7 shows the number (and

percentage) of unique AS peers that establish certain types of peerings, along with the number (and percentage) of corresponding *CBIs* and *ABIs* for those peers. Since there are overlapping ASes and interfaces between different groups, Table 7 also presents three rows (i.e. rows 3, 6, and 9 with italic fonts) that aggregate the information for the two closely related prior pair of rows/groups. These three aggregate rows provide an overall view of Amazon's inferred peering fabric that highlight two points of general interest: *(i)* While 76% of Amazon's peers use Pb peering, only 33% of Amazon's peers use Pr-nB (virtual or physical) peerings, with the overlap of about 10% of peer ASes relying on both Pr-nB and Pb peerings, and the fraction of Pr-B peerings being very small (3%). *(ii)* The average number of *CBIs* (and *ABIs*) for ASes that use Pr-B, Pr-nB and Pb peerings to interconnect with Amazon is 65 (17), 11 (2), and 2 (0.3), respectively.

Hidden Peerings. Note that there are groups of Amazon's inferred peerings shown in Table 7 (together with their associated traffic) that remain in general hidden from the measurement techniques that are commonly used for inferring peerings (e.g. traceroute). One such group consists of all the virtual peerings (Pr-*-V) since they are used to exchange traffic between customer ASes of Amazon (or their downstream ASes) and Amazon. The second group is made up of all other non-virtual peerings that are not visible in BGP data, namely Pr-nB-nV and even Pb-nB. The presence of these peerings cannot be inferred from public BGP data and their associated traffic is only visible along the short AS path to the customer AS. These hidden peerings make up 33.29% of all of Amazon's inferred peerings and their associated traffic is carried over Amazon's private backbone and not over the public Internet.

Table 8. Hybrid peering groups along with the number of unique ASes for each group.

Different Types of Hybrid Peering	**#ASN**
Pb-nB	2187
Pr-nB-nV	686
Pr-nB-nV; Pb-nB	207
Pb-B	117
Pr-nB-nV; Pr-nB-V	83
Pr-nB-nV; Pb-nB; Pr-nB-V	60
Pb-nB; Pr-nB-V	41
Pr-nB-V	38
Pr-B-nV; Pb-B	37
Pr-B-V; Pr-B-nV; Pb-B	31
Pr-B-nV	24
Pr-B-V; Pr-B-nV	16
Pr-nB-nV; Pr-B-nV; Pr-B-V	5
Pr-B-V; Pb-B	4
Pr-B-V	4
Pb-nB; Pb-B	2
Pr-nB-nV; Pr-B-nV; Pr-B-V; Pb-B	2
Pr-nB-nV; Pr-B-nV	1
Pr-nB-nV; Pr-B-nV; Pb-B	1
Pr-nB-nV; Pr-nB-V; Pr-B-nV	1
Pr-nB-nV; Pr-nB-V; Pr-B-nV; Pr-B-V; Pb-B	1

Hybrid Peering. Individual ASes may establish multiple peerings of different types (referred to as “hybrid" peering) with Amazon; that is, appear as a member of two (or more) groups in Table 7. We group all ASes that establish such hybrid peering based on the combination of peering types that are listed in Table 7 types and that they maintain with Amazon. The following are two of the most common hybrid peering scenarios we observe. **Pr-nB-nV + Pb-nB:** With 207 ASes, this is the largest group of ASes which utilize hybrid peering. Members of this group use both types of peerings to exchange their own traffic with Amazon and include ASes such as Akamai, Intercloud, Datapipe, Cloudnet, and Dell. **Pr-nB-nV; Pb-nB; Pr-nB-V:** This group is similar to the first group one but its members also utilize

virtual peerings to exchange their own traffic with Amazon. This group consists of 60 ASes that include large providers such as Google, Microsoft, Facebook, and Limelight. Table 8 gives a detailed breakdown of the observed hybrid (and non-hybrid) peering groups and shows for each group the number of ASes that use that peering group. Note that each AS is counted only once in the group that has the most specific peering types.

4.7.3 Inferring the Purpose of Peerings. In an attempt to gain insight into how each of the six different groups of Amazon's peerings is being used in practice, we consider a number of additional characteristics of the peers in each group and depict those characteristics using stacked boxplots as shown in Figure 19. In particular, starting with the top row in Figure 19, we consider summary distributions of [11] *(i) size of customer cone of peering AS* (i.e. number of /24 prefixes that are reachable through the AS (labeled as "BGP /24"); *(ii)* number of /24 prefixes that are reachable from Amazon through the identified *CBIs* associated with each peering; *(iii)* number of *ABIs* for individual peering AS; *(iv)* number of *CBIs* for individual peering AS; *(v)* min RTT difference between both ends of individual peering; *(vi)* number of unique metro areas that the *CBIs* of each peering AS have been pinned to (see § 4.6).

For example, we view the number of /24 prefixes in the customer cone of an AS to reflect the AS's size/role (i.e. as tier-1 or tier-2 AS) in routing Internet traffic. Moreover, comparing the number of /24 prefixes in the customer cone with the number of reachable /24 prefixes through a specific peering for an AS reveals the purpose of the corresponding peering to route traffic to/from Amazon from/to its downstream networks. In the following, we discuss how the combined

[11] For ASes that utilize hybrid peering with Amazon, the reported information in each group only includes peerings related to that group.

information in Table 7 and Figure 19 sheds light on Amazon's global-scale peering fabric and illuminates the different roles of the six groups of peering ASes.

Pb-nB. The peers in this group are typically edge networks with a small customer cone (including content, enterprise, and smaller transit/access networks) that exchange traffic with Amazon through a single *CBI* at an IXP. The corresponding routes are between Amazon and these edge networks and are thus *not* announced in BGP. Peers in this group include CDNs like Akamai, small transit/access providers like Etisalat, BT, and Floridanet, and enterprises such as Adobe, Cloudflare, Datapipe (Rackspace), Google, Symantec, LinkedIn, and Yandex.

Pb-B. This group consists mostly of tier-2 transit networks with moderate-sized customer cones. These networks are present at a number of IXPs to connect their their downstream customer networks to Amazon. The corresponding routes must be announced to downstream ASes and are thus visible in BGP. Example peers in this group are CW, DigitalOcean, Fastweb, Seabone, Shaw Cable, Google Fiber, and Vodafone.

Pr-nB-V. The peers in this group are a combination of small transit providers and some content and enterprise networks. They establish VPIs at a single location to exchange either their own traffic or the traffic of their downstream networks with Amazon through a VPI. Therefore, their peering is not visible in BGP. About 85% of these peers are visible from two cloud providers while the rest is visible from more than two cloud providers. Examples of enterprise and content networks in this group are Apple, UCSD, UIOWA, LG, and Edgecast, and examples of transit networks are Rogers, Charter, and CenturyLink.

Pr-nB-nV. These peers appear to establish physical interconnections (i.e. cross-connects) with Amazon since they are not reachable from other cloud providers.

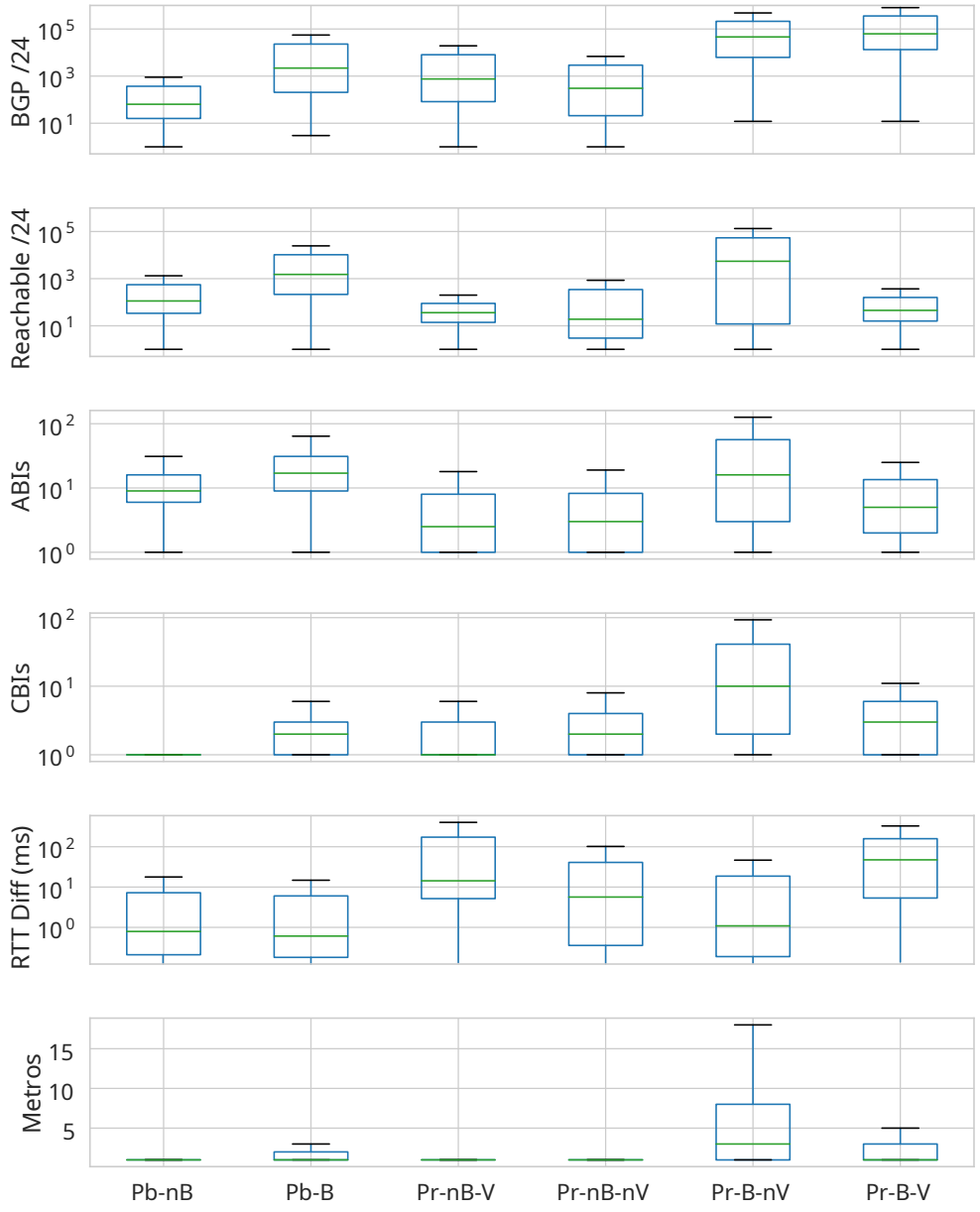


Figure 19. Key features of the six groups of Amazon's peerings (presented in Table 7) showing (from top to bottom): the number of /24 prefixes within the customer cone of peering AS, the number of probed /24 prefixes that are reachable through the *CBIs* of associated peerings of an AS, the number of *ABIs* and *CBIs* of associated of an AS, the difference in RTT of both ends of associated peerings of an AS, and the number of metro areas which the *CBIs* of each peering AS have been pinned to.

However, given the earlier-mentioned under-counting of VPIs by our method,

we hypothesize that some or all of these peerings could be associated with VPIs,

similar to the previous group. The composition of the peers in this group is comparable to **Pr-nB-V** but includes a larger fraction of enterprise networks (i.e. main users of VPIs) which in turn is consistent with our hypothesis. Examples of peers in this group are enterprises such as Datapipe (Rackspace), Chevron, Vox-Media, UToronto, and Georgia-Tech, CDNs such as Akamai and Limelight and transit/access providers like Comcast. To further examine our hypothesis, we parse the DNS names of 4.85k *CBIs* associated with peers in the **Pr-nB** group. 170 of these DNS names (100 from Pr-nB-nV and 70 from Pr-nB-V interfaces) contain VLAN tags, indicating the presence of a virtual private interconnection. We also observe some commonly used (albeit not required) keywords Amazon (2018e) such as *dxvif* (Amazon terminology for "direct connect virtual interface"), *dxcon, awsdx* and *aws-dx* for 125 (out of 170) *CBIs* where the "dx"-notation is synonymous with an interface's use for "direct interconnections". We consider the appearance of these keywords in the DNS names of *CBIs* for this group of peerings (and only in this group) as strong evidence that the interconnections in question are indeed VPIs. Therefore, a subset of Pr-nB-nV interconnections is likely to be virtual as well.

Pr-B-nV. The peers in this group are very large transit networks that establish cross-connections at various locations (many *CBIs* and *ABIs*) across the world). The large number of prefixes that are reachable through them from Amazon and the visibility of the peerings in BGP suggest that these peers simply provide connectivity for their downstream clients to Amazon. Given the large size of these transit networks, the visibility of these peerings in BGP is due to the announcement of routes from Amazon to all of their downstream networks. Intuitively, given the volume of aggregate traffic exchanged between Amazon and these large transit networks, the peers in this group have the largest number

of *CBIs*, and these *CBIs* are located at different metro areas across the world. Example networks in this group are AT&T, Level3 (now CenturyLink), GTT, Cogent, HE, XO, Zayo, and NTT.

Pr-B-V. This group consists mostly a subset of the very large transit networks in **Pr-B-nV** and the peers in this group also establish a few VPIs (at different locations) with Amazon. The small number of prefixes that are reachable from Amazon through these peers along with the large number of *CBIs* per peer indicates that these peers bring specific Amazon clients (a provider or enterprise, perhaps even without an ASN) to a colo facility to exchange traffic with Amazon Amazon (2018c). The presence of these peerings in BGP is due to the role they play as transit networks in the **Pr-B-nV** group that is separate from peers in this group using virtual peerings. Example networks in this group are Cogent, Comcast, CW, GTT, CenturyLink, HE, and TimeWarner, all of which are listed as Amazon cloud connectivity partners Amazon (2018c); Google (2018c); Microsoft (2018b)) and connect enterprises to Amazon. When examining the min RTT difference between both ends of peerings across different groups (row 5 in Figure 19), we observe that both groups with virtual interconnections (Pr-B-V and Pr-nB-V) have in general larger values than the other groups. This observation is in agreement with the fact that many of these VPIs are associated with enterprises that are brought to the cloud exchange by access networks using layer-2 connections.

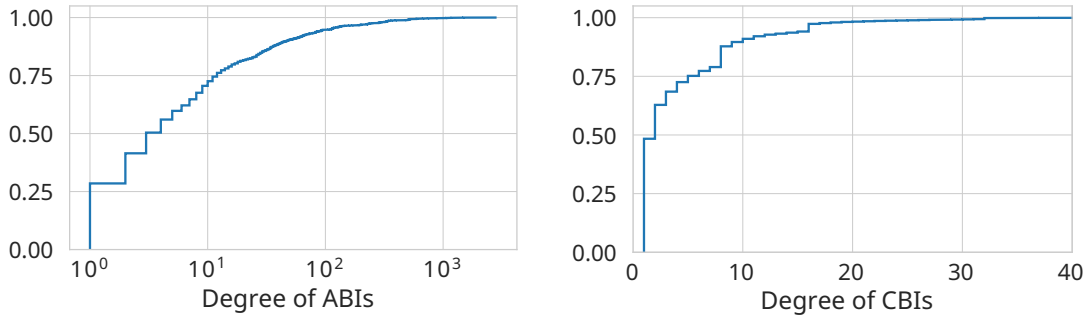
Coverage of Amazon's Interconnections. Although the total number of peerings that Amazon has with its customers is not known, our goal here is to provide a baseline comparison between Amazon's peering fabric that is visible in public BGP data and Amazon's peering fabric as inferred by our approach. Using our approach, we have identified 3.3k unique peerings for Amazon. In contrast,

there are only 250 unique Amazon peerings reported in BGP, and 226 of them are also discovered by our approach. Upon closer examination, for some of the 24 peerings that are seen in BGP but not by our approach, we observed a sibling of the corresponding peer ASes. This brings the total coverage of our method to about 93% of all reported Amazon peerings in BGP. In addition, we report on more than 3k unique Amazon peerings that are not visible in public BGP data. These peerings with Amazon and their associated traffic are not visible when relying on more conventional measurement techniques.

4.7.4 Characterizing Amazon's Connectivity Graph. Having focused so far on groups of peerings of certain types or individual AS peers, we next provide a more holistic view of Amazon's inferred peering graph and examine some of its basic characteristics. We first produce the Interface Connectivity Graph (ICG) between all the inferred border interfaces. ICG is a bipartite graph where each node is a border interface (an *ABI* or a *CBI*) and each edge corresponds to the traceroute interconnection segment (ICS) between an *ABI* and a *CBI*. We also annotate each edge with the difference in the minimum RTT from the closest VM to each end of the ICS.[12]

Intuitively, we expect the resulting ICG to have a separate partition that consists of interconnections associated with each region, i.e. *ABIs* of a region connecting to *CBIs* that are supported by them. However, we observe that the ICG's largest connected component consists of the vast majority (92.3%) of all nodes. This implies that there are links between *ABIs* in each Amazon region and *CBIs* in several other regions. Upon closer examination of 57.85% of all the peerings that have both of their ends pinned, we notice that a majority of these

[12]We identify the VM that has the shortest RTT from an *ABI* and use the min-RTT of the same VM from the corresponding *CBI* to determine the RTT of an ICS.



(a) (b)

Figure 20. Distribution of *ABIs* (log scale) and *CBIs* degree in left and right figures accordingly.

peerings (98%) are indeed contained within individual Amazon regions. However, we do encounter remote peerings between regions that are a significant geographical distance apart. For example, there are peerings between FR and KR, US-VA and SG, AU and CA. The large fraction of peerings with only one end or no end pinned (about 42%) suggests that the actual number of remote peerings is likely to be much larger. These remote peerings are the main reason for why the ICG's largest connected component contains more than 92% of all border interfaces.

To illustrate the basic connectivity features of the bi-partite ICG, Figures 20a and 20b show the distributions of the number of *CBIs* that are associated with each individual *ABIs* (degree of *ABIs*) and the number of *ABIs* associated with individual *CBIs* (degree of *CBIs*). We observe a skewed distribution for *ABI* degree where 30%, 70%, and 95% of *ABIs* are associated with 1, <10, and <100 *CBIs*, respectively. Roughly 50% (90%) of *CBIs* are associated with a single (≤ 8) *ABIs*. A closer examination shows that high degree *CBIs* are mainly associated with Amazon's public peerings with large transit networks (e.g. GTT, Cogent, NTT, CenturyLink). In contrast, a majority of high degree *ABIs* is associated with private, non-BGP, non-virtual peerings (see § 4.7).

4.8 Inferring Peering with bdrmap

As stated earlier in § 4.2, *bdrmap* Luckie et al. (2016)[13] is the only other existing tool for inferring border routers of a given network from traceroute data. With Amazon as the network of interest, our setting appears to be a perfect fit for the type of target settings assumed by *bdrmap*. However, there are two important differences between the cloud service provider networks we are interested in (e.g. Amazon) and the more traditional service provider network that *bdrmap* targets (e.g. a large US Tier-1 network). First, not only can the visibility of different prefixes vary widely across different Amazon regions, but roughly one-third of Amazon's peerings are not visible in BGP and even some of the BGP-visible peerings of a network are related to other instances of its peerings with Amazon (§ 4.7). At the same time, *bdrmap* relies on peering relationships in BGP to determine the targets for its traceroute probes and also uses them as input for some of its heuristics. Therefore, *bdrmap*'s outcome is affected by any inconsistent or missing peering relationship in BGP. Second, as noted earlier, our traceroute probes reveal hybrid Amazon border routers that have both Amazon and client routers as their next hop and connect to them. This setting is not consistent with *bdrmap*'s assumption that border routers should be situated exclusively in the host or peering network. Given these differences, the comparison below is intended as a guideline for how *bdrmap* could be improved to apply in a cloud-centric setting.

Thanks to special efforts by the authors of bdrmap who modified their tool so it could be used for launching traceroutes from cloud-based vantage points (i.e., VMs), we were able to run it in all Amazon regions to compare the bdrmap-inferred border routers with our inference results. *bdrmap* identified 4.83k *ABIs*

[13] *MAP-IT* Marder and Smith (2016) and *bdrmapIT* Alexander et al. (2018) are not suitable for this setting since we have layer-2 devices at the border.

and 9.65k *CBIs* associated with 2.66k ASes from all global regions. 3.23k of these *CBIs* belong to IXP prefixes and are associated with 1.81k ASes. Given *bdrmap*'s customized probing strategy and its extensive use of different heuristics, it is not feasible to identify the exact reasons for all the observed differences between *bdrmap*'s and our findings. However, we were able to identify the following three major inconsistencies in *bdrmap*'s output.

First, *bdrmap* does not report an AS owner for 0.32k of its inferred *CBIs* (i.e. owner is AS0). Second, instances of *bdrmap* that run in different Amazon regions report different AS owners for more than 500 *CBIs*, sometimes as many as 4 or 5 different AS owners for an interface. Third, running instances of *bdrmap* in different Amazon regions results in inconsistent views of individual border router interfaces; e.g. one and the same interface is inferred to be an *ABI* from one region and a *CBI* from another region. We identified 872 interfaces that exhibit this inconsistency. Furthermore, the fact that 97% (846 out of 872) of the interfaces with this type of inconsistency are advertised by Amazon's ASNs indicates that the AS owner for these interfaces have been inferred by *bdrmap*'s heuristics.

When comparing the findings of *bdrmap* against our methodology in more detail, we observed that our methodology and *bdrmap* have 1.85k, 5.48k, and 2k *ABI*, *CBI*, and ASes in common. However, without access to ground truth, a full investigation into the various points of disagreement is problematic. To make the problem more tractable, we limit our investigation to the 0.65k ASes that were exclusively identified by *bdrmap* and try to rely on other sources of information to confirm or dismiss *bdrmap*'s findings. These exclusive ASNs belong to 0.18k (0.49k) IXP (private) peerings. For IXP peerings, we compare *bdrmap*'s findings against IP-to-ASN mappings that are published by IXP operators or rely on embedded

information within DNS names. The inferences of *bdrmap* is only aligned for 42 of these peers. For the 0.49k private peerings we focus on inferences that were made by the *thirdparty* heuristic as it constitutes the largest (62%) fraction of *bdrmap*-exclusive private peerings (for details, see § 5.4 in Luckie et al. (2016)). These ASes are associated with 375 *CBIs* and we observe 66 (60 ASNs) of these interfaces in our data. For each of these 66 *CBIs*, we calculate the set of reachable destination ASNs through these *CBIs* and determine the upstream provider network for each one of these destination ASes using BGP data CAIDA (2018). Observing more than one or no common provider network among reachable destination ASes for individual *CBIs* would invalidate the application of *bdrmap*'s *thirdparty* heuristic, i.e. *bdrmap* wouldn't have applied this heuristic if it had done more extensive probing that revealed an additional set of reachable destination ASes for these *CBIs*. We find that 50 (44 ASNs) out of the 66 common *CBIs* have more than one or no common providers for the target ASNs. Note that this observation does not invalidate *bdrmap*'s *thirdparty* heuristics but highlights its reliance on high-quality BGP snapshots and AS-relationship information.

4.9 Limitations of Our Study

As a third-party measurement study of Amazon's peering fabric that makes no use of Amazon-proprietary data and only relies on generally-available measurement techniques, there are inherent limitations to our efforts aimed at inferring and geo-locating all interconnections between Amazon and the rest of the Internet. This section collects and organizes the key limitations in one place and details their impact on our findings.

Inferring Interconnections. Border routers responding to traceroute probes using a third-party address are a well-known cause for artifacts in traceroute

measurement output, and our IXP-client and Hybrid-IP heuristics used in § 4.5.1 are not immune to this problem. However, as reported in Luckie et al. (2014), the fraction of routers that respond with their incoming interface is in general above 50% and typically even higher in the U.S.

In contrast, because of the isolation of network paths for VPIs of Amazon's clients that use private addresses, any peerings associated with these VPIs are not visible to probes from VMs owned by other Amazon customers. As a result, our inference methodology described in § 4.4 cannot discover established VPIs that leverage private IP addresses.

Pinning Interconnections. In § 4.6, we reported being able to pin only about half of all the inferred peering interfaces at the metro level. In an attempt to understand what is limiting our ability to pin the rest of the inferred interfaces, we identified two main reasons. First, there is a lack of anchors in certain regions, and second, there is the common use of remote peering. These two factors in conjunction with our conservative iterative strategy for pinning interfaces to the metro level make it difficult to provide enough and sufficiently reliable indicators of interface-specific locations.

One way to overcome some of these limiting factors is by using a coarser scale for pinning (e.g. regional level). In fact, as shown in § 4.6, at the regional level, we are able to pin some 30% of the remaining interfaces which improves the overall coverage of our pinning strategy at the granularity of regions to about 80%.

Other Observations. Although our study does not consider IPv6 addresses, we argue that the proposed methodology only requires minimal modifications (e.g. incorporating IPv6 target selection techniques Beverly et al. (2018); Gasser et al.

(2018)) to be applicable to infer IPv6 peerings. We will explore IPv6 peerings as part of future work.

Like others before us, as third-party researchers, we found it challenging to validate our Amazon-specific findings. Like most of the large commercial provider networks, Amazon makes little, if any, ground truth data about its global-scale serving infrastructure publicly available, and our attempts at obtaining peering-related ground truth information from either Amazon, Amazon's customers, operators of colo facilities where Amazon is native, or AWS Direct Connect Partners have been futile.

Faced with the reality of a dearth of ground truth data, whenever possible, we relied on extensive consistency-checking of our results (e.g. see § 4.5, § 4.6). At the same time, many of our heuristics are conservative in nature, typically requiring agreement when provided with input from multiple complementary sources of information. As a result, the reported quantities in this chapter are in general lower bounds but nevertheless demonstrate the existence of a substantial number of Amazon-related peerings that are not visible to more conventional measurement studies and/or inference techniques.

4.10 Summary

In this chapter, we present a measurement study of the interconnection fabric that Amazon utilizes on a global scale to run its various businesses, including AWS. We show that in addition to some 0.12k private peerings and about 2.69k pubic peerings (i.e., bi-lateral and multi-lateral peerings), Amazon also utilizes at least 0.24k (and likely many more) virtual private interconnections or VPIs. VPIs are a new and increasingly popular interconnection option for entities such as enterprises that desire highly elastic and flexible connections to the cloud providers

that offer the type of services that these entities deem critical for running their business. Our study makes no use of Amazon-proprietary data and can be used to map the interconnection fabric of any large cloud provider, provided the provider in question does not filter traceroute probes.

Our findings emphasize that new methods are needed to track and study the type of "hybrid" connectivity that is in use today at the Internet's edge. This hybrid connectivity describes an emerging strategy whereby one part of an Internet player's traffic bypasses the public Internet (i.e. cloud service-related traffic traversing cloud exchange-provided VPIs), another part is handled by its upstream ISP (i.e. traversing colo-provided private interconnections), and yet another portion of its traffic is exchanged over a colo-owned and colo-operated IXP. As the number of businesses investing in cloud services is expected to continue to increase rapidly, multi-cloud strategies are predicted to become mainstream, and the majority of future workload-related traffic is anticipated to be handled by cloud-enabled colos Gartner (2016), tracking and studying this hybrid connectivity will require significant research efforts on parts of the networking community. Knowing the structure of this hybrid connectivity, for instance, is a prerequisite for studying which types of interconnections will handle the bulk of tomorrow's Internet traffic, and how much of that traffic will bypass the public Internet, with implications on the role that traditional players such as Internet transit providers and emerging players such as cloud-centric data center providers may play in the future Internet.

CHAPTER V

CLOUD CONNECTIVITY PERFORMANCE

5.1 Introduction

In Chapter IV we presented and characterized different peering relationships that CPs form with various networks. This chapter focuses on the performance of various connectivity options that are at the disposal of enterprises for establishing end-to-end connectivity with cloud resources.

The content in this chapter is the result of a collaboration between Bahador Yeganeh with Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. Bahador Yeganeh is the primary author of this work and responsible for conducting all measurements and producing the presented analyses.

5.2 Introduction

For enterprises, the premise of deploying a multi-cloud strategy[1] is succinctly captured by the phrase “not all clouds are equal". That is, instead of considering and consuming compute resources as a utility from a single cloud provider (CP), to better satisfy their specific requirements, enterprise networks can pick-and-choose services from multiple participating CPs (e.g. rent storage from one CP, compute resources from another) and establish end-to-end connectivity between them and their on-premises server(s) at the same or different locations. In the process, they also avoid vendor lock-in, enhance the reliability and performance of the selected services, and can reduce the operational cost of deployments. Indeed, according to an industry report from late 2018 Krishna et al. (2018), 85% of the enterprises have already adopted multi-cloud strategies, and that number is expected to rise to 98% by 2021. Because of their popularity with enterprise networks, multi-

[1]This is different from hybrid cloud computing, where a direct connection exists between a public cloud and private on-premises enterprise server(s).

cloud strategies are here to stay and can be expected to be one of the drivers of innovation in the future cloud services *The enterprise deployment game-plan: why multi-cloud is the future* (2018); *Five Reasons Why Multi-Cloud Infrastructure is the Future of Enterprise IT* (2018); *The Future of IT Transformation Is Multi-Cloud* (2018); *The Future of Multi-Cloud: Common APIs Across Public and Private Clouds* (2018); *The Future of the Datacenter is Multicloud* (2018); *How multi-cloud business models will shape the future* (2018); *IBM bets on a multi-cloud future* (2018).

Fueled by the deployment of multi-cloud strategies, we are witnessing two new trends in Internet connectivity. First, there is the emergence of new Internet players in the form of third-party private connectivity providers (e.g. DataPipe, HopOne, among others Amazon (2018c); Google (2018b); Microsoft (2018c)). These entities offer direct, secure, private, layer 3 connectivity between CPs (henceforth referred to as *third-party private* (TPP)), at a cost of a few hundreds of dollars per month. TPP routes bypass the public Internet at Cloud Exchanges CoreSite (2018); Demchenko et al. (2013) and offer additional benefits to users (e.g. enterprise networks can connect to CPs without owning an Autonomous System Number, or ASN, or physical infrastructure). Second, the large CPs are aggressively expanding the footprint of their serving infrastructures, including the number of direct connect locations where enterprises can reach the cloud via direct, private connectivity (henceforth referred to as *cloud-provider private* (CPP)) using either new CP-specific interconnection services (e.g. Amazon (2018a); Google (2018a); Microsoft (2018a)) or third-party private connectivity providers at colocation facilities. Of course, a user can forgo the TPP and CPP options altogether and rely instead on the traditional, best-effort connectivity over the

public Internet—henceforth referred to as *(transit provider-based) best-effort public (Internet)* (BEP)—to employ a multi-cloud strategy.

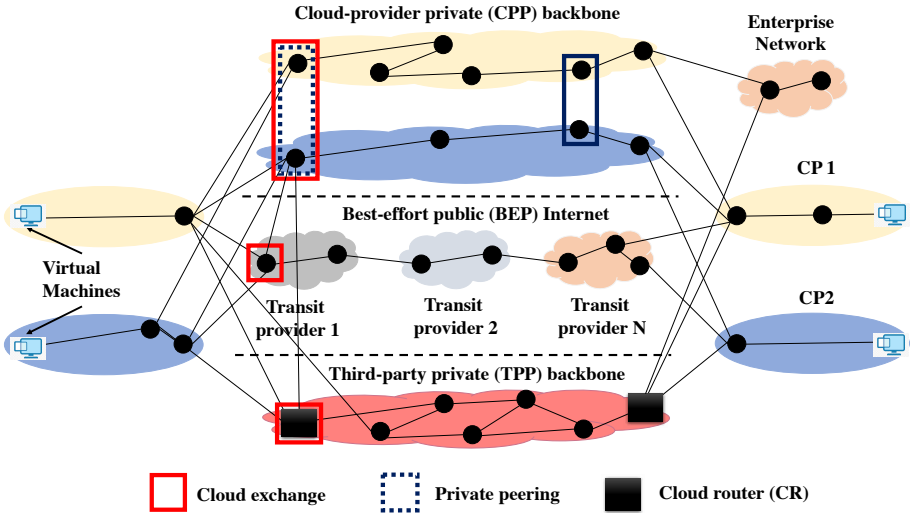


Figure 21. Three different multi-cloud connectivity options.

To illustrate the problem, consider, for example, the case of a modern enterprise whose goal is to adopt a multi-cloud strategy (i.e. establishing end-to-end connectivity between (i) two or more CPs, i.e. cloud-to-cloud; and (ii) enterprise servers and the participating CPs, i.e. enterprise-to-cloud) that is performance- and cost-aware. For this scenario, let us assume that (a) the enterprise's customers are geo-dispersed and different CPs are available in different geographic regions (i.e. latency matters for all customers); (b) regulations are in place (e.g. for file sharing and storing data in EU; hence, throughput matters for data transfers *Example Applications Services* (2018)); (c) cloud reliability and disaster recovery are important, especially in the face of path failures (i.e. routing matters); and (d) cost savings play an important role in connectivity decisions. Given these requirements, the diversity of CPs, the above-mentioned different connectivity options, and the lack of visibility into the performance tradeoffs, routing choices, and topological features associated with these multi-cloud connectivity options, the enterprise faces

the "problem of plenty": *how to best leverage the different CPs' infrastructures, the various available connectivity choices, and the possible routing options to deploy a multi-cloud strategy that achieves the enterprise's performance and cost objectives?*

With multi-cloud connectivity being the main focus of this chapter, we note that existing measurement techniques are a poor match in this context. For one, they fall short of providing the data needed to infer the type of connectivity (i.e. TPP, CPP, and BEP) between (two or more) participating CPs. Second, they are largely incapable of providing the visibility needed to study the topological properties, performance differences, or routing strategies associated with different connectivity options. Last but not least, while mapping the connectivity from cloud/content providers to users has been considered in prior work (e.g. Anwar et al. (2015); Calder, Flavel, Katz-Bassett, Mahajan, and Padhye (2015); Calder et al. (2018); Chiu et al. (2015); Cunha et al. (2016); Schlinker et al. (2017) and references therein), multi-cloud connectivity from a cloud-to-cloud (C2C) perspective has remained largely unexplored to date.

This chapter aims to empirically examine the different types of multi-cloud connectivity options that are available in today's Internet and investigate their performance characteristics using non-proprietary cloud-centric, active measurements. In the process, we are also interested in attributing the observed characteristics to aspects related to connectivity, routing strategy, or the presence of any performance bottlenecks. To study multi-cloud connectivity from a C2C perspective, we deploy and interconnect VMs hosted within and across two different geographic regions or availability zones (i.e. CA and VA) of three large cloud providers (i.e. Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure) using the TPP, CPP, and BEP option, respectively. We note that

the high cost of using the services of commercial third-party private connectivity providers for implementing the TPP option prevents us from having a more global-scale deployment that utilizes more than one such provider.

Using this experimental setup as a starting point, we first compare the stability and/or variability in performance across the three connectivity options using metrics such as delay, throughput, and loss rate over time. We find that CPP routes exhibit lower latency and are more stable when compared to BEP and TPP routes. CPP routes also have higher throughput and exhibit less variation compared to the other two options. Given that using the TPP option is expensive, this finding is puzzling. In our attempt to explain this observation, we find that inconsistencies in performance characteristics are caused by several factors including border routers, queuing delays, and higher loss-rates of TPP routes. Moreover, we attribute the CPP routes' overall superior performance to the fact that each of the CPs has a private optical backbone, there exists rich inter-CP connectivity, and that the CPs' traffic *always* bypasses (i.e. is invisible to) BEP transits.

In summary, this chapter makes the following contributions:

- To the best of our knowledge, this is one of the first efforts to perform a comparative characterization of multi-cloud connectivity in today's Internet. To facilitate independent validation of our results, we will release all relevant datasets (properly anonymized; e.g. with all TPP-related information removed).
- We identify issues, differences, and tradeoffs associated with three popular multi-cloud connectivity options and strive to elucidate/discuss the underlying reasons. Our results highlight the critical need for open measurement platforms and more transparency by the multi-cloud connectivity providers.

The rest of the chapter is organized as follows. We describe the measurement framework, cloud providers, performance metrics, and data collection in § 5.3. Our measurements results and root causes, both from C2C and E2C perspectives are in § 5.4 and § 5.5 respectively. We present the open issues and future work in § 5.6. Finally, we summarize the key findings of this chapter in § 5.7.

5.3 Measurement Methodology

In this section, we describe our measurement setting and how we examine the various multi-cloud connectivity options, the cloud providers under consideration, and the performance metrics of interest.

5.3.1 Deployment Strategy. As shown in Figure 21, we explore in this chapter three different types of multi-cloud connectivity options: *third-party private* (TPP) connectivity between CP VMs that bypasses the public Internet, *cloud-provider private* (CPP) connectivity enabled by private peering between the CPs, and *best-effort public* (BEP) connectivity via transit providers. To establish TPPs, we identify the set of colocation facilities where connectivity partners offer their services Amazon (2018c); Google (2018b); Microsoft (2018c). Using this information, we select colocation facilities of interest (e.g. in the geo-proximity of cloud VMs) and deploy the third-party providers' cloud routers (CRs) that interconnect virtual private cloud networks within a region or regions. The selection of CR locations can also leverage latency information obtained from the third-party connectivity providers.

Next, based on the set of selected VMs and CRs we utilize third-party connectivity APIs to deploy CRs and establish virtual cloud interconnections between VMs and CRs to create TPPs. At a high level, this step involves (i)

establishing a virtual circuit between the CP and a connectivity partner, (ii) establishing a BGP peering session between the CP's border routers and the partner's CR, (iii) connecting the virtual private cloud gateway to the CP's border routers, and (iv) configuring each cloud instance to route any traffic destined to the overlay network towards the configured virtual gateway. Establishing CPP connectivity is similar to TPP. The only difference is in the user-specified connectivity graph where in the case of CPP, CR information is omitted. To establish CPP connectivity, participating CPs automatically select private peering locations to stitch the multi-cloud VMs together. Finally, we have two measurement settings for BEP. While the first setting is between a non-native colocation facility in AZ and our VMs through the BEP Internet, the second form of measurement is towards Looking Glasses (LGs) residing in the colocation facility hosting our CRs, also traverses the BEP Internet, and only yields latency measurements.

Our network measurements are performed in rounds. Each round consists of path, latency, and throughput measurements between all pairs of VMs (in both directions to account for route asymmetry) but can be expanded to include additional measurements as well. Furthermore, the measurements are performed over the public BEPs as well as the two private options (i.e. CPP and TPP). We avoid cross-measurement interference by tracking the current state of ongoing measurements and limit measurement activities to one active measurement per cloud VM. The results of the measurements are stored locally on the VMs (hard disks) and are transmitted to a centralized storage at the end of our measurement period.

5.3.2 Measurement Scenario & Cloud Providers.

As mentioned earlier, the measurement setting is designed to provide visibility into multi-cloud deployments so as to be able to study aspects related to the topology, routing, and performance tradeoffs. Unfortunately, the availability of several TPP providers, and more importantly, the incurred costs for connecting multiple clouds using TPP connections are very high. For example, for each 1 Gbps link to a CP network, third party providers charge anywhere from about 300 to 700 USD per month Megaport (2019a); PacketFabric (2019); Pureport (2019)[2]. Such high costs of TPP connections prevents us from having a global-scale deployment and the possibility of examining multiple TPP providers. Due to the costly nature of establishing TPP connections, we empirically measure and examine only one coast-to-coast, multi-cloud deployment in the US. The deployment we consider in this study is nevertheless representative of a typical multi-cloud strategy that is adopted by modern enterprises Megaport (2019b).

More specifically, our study focuses on connectivity between three major CPs (AWS, Azure, and GCP) and one enterprise. To emulate realistic multi-cloud scenarios, each entity is associated with a geographic location.The deployments are shown in Figure 22. We select the three CPs as they collectively have a significant market share and are used by many clients concurrently ZDNet (2019). Using these CPs, we create a realistic multi-cloud scenario by deploying three CRs using one of the top third-party connectivity provider's network; one in the Santa Clara, CA (CR-CA) region, one in the Phoenix, AZ (CR-AZ) region, and one in the Ashburn, VA (CR-VA) region. CR-CA is interconnected to CR-VA and CR-AZ. Furthermore, CR-CA and CR-VA are interconnected with native cloud VMs from Amazon,

[2]Note that these price points do not take into consideration the additional charges that are incurred by CPs for establishing connectivity to their network.

Google, and Microsoft. To emulate an enterprise leveraging the multi-clouds, CR-AZ is connected to a physical server hosted within a colocation facility in Phoenix, AZ (server-AZ).

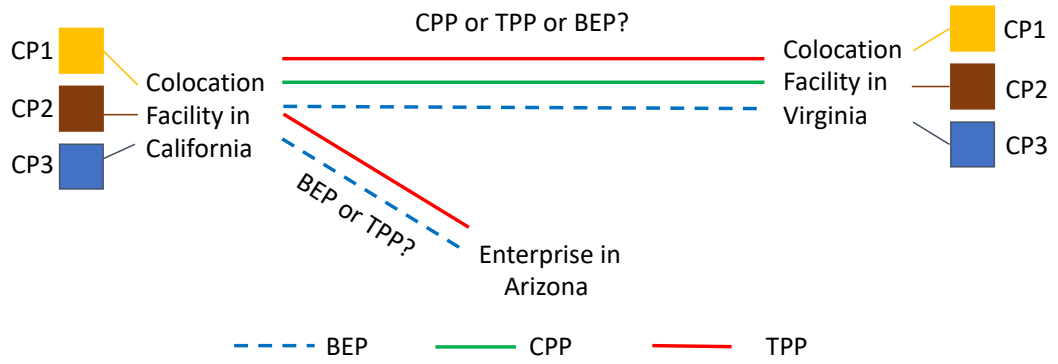


Figure 22. Our measurement setup showing the locations of our VMs from AWS, GCP and Azure. A third-party provider's CRs and line-of-sight links for TPP, BEP, and CPP are also shown.

The cloud VMs and server-AZ are all connected to CRs with 50Mb/s links. We select the colocation facility hosting the CRs based on two criteria (i) CPs offer native cloud connectivity within that colo, and (ii) geo-proximity to the target CPs datacenters. CRs are interconnected with each other using a 150Mb/s link capacity that support the maximum (3 concurrent measurements in total to avoid more than 1 ongoing measurement per VM) number of concurrent measurements that we perform. Each cloud VM has at least 2 vCPU cores, 4GB of memory, and runs Ubuntu server 18.04 LTS. Our VMs were purposefully over-provisioned to reduce any measurement noise within virtualized environments. Throughout our measurements the VMs CPU utilization always remained below 2%. We also cap the VM interfaces at 50Mb/s to have a consistent measurement setting for both public (BEP) and private (TPP and CPP) routes. We perform measurements between all CP VMs within regions (intra-region), across regions (inter-region) for C2C analysis, and from server-AZ to VMs in CA for E2C analysis. Additionally,

we also perform measurements between our cloud VMs and two LGs that are located within the same facility as CR-CA and CR-VA, respectively, and use these measurements as baselines for comparisons (C2LG). Together, these efforts resulted in 60 pairs of measurements between CP instances ($P(6,2) * 2$ permutation of 2 pairs out of 6 CP VMs over 2 types of unidirectional network paths), 24 pairs of measurement between CP VMs and LGs (6 CP VMs * 2 LGs * 2 type of unidirectional network paths), and 12 pairs of measurement between server-AZ and west coast CP VMs ($P(3,2) * 2$ permutation of 3 west coast CP VMs over 2 types of unidirectional network paths).

5.3.3 Data Collection & Performance Metrics. Using our measurement setting, we conducted measurements for about a month in the Spring of 2019.[3] We conduct measurements in 10-minute rounds. In each round, we performed latency, path, and throughput measurements between all pairs of relevant nodes. For each round, we measure and report the latency using 10 *ping* probes. We refrain from using a more accurate one-way latency measurement tool such as OWAMP as the authors of OWAMP caution its use within virtualized environments *One-Way Ping (OWAMP)* (2019). Similarly, paths are measured by performing 10 attempts of *paris-traceroute* using *scamper* Luckie (2010) towards each destination. We used ICMP probes for path discovery as they maximized the number of responsive hops along the forward path. Lastly, throughput is measured using the *iperf3* tool, which was configured to transmit data over a 10-second interval using TCP. We discard the first 5 seconds of our throughput measurement to account for TCP's *slow-start* phase and consider the median of throughput for

[3]See § 5.3.5 for more details.

the remaining 5 seconds. These efforts resulted in about 30k latency and path samples and some 15k throughput samples between each measurement pair.

To infer inter-AS interconnections, the resulting traceroute hops from our measurements were translated to their corresponding AS paths using BGP prefix announcements from Routeviews and RIPE RIS RIPE (2019); University of Oregon (2018). Missing hops were attributed to their surrounding ASN if the prior and next hop ASNs were identical. The existence of IXP hops along the forward path was detected by matching hop addresses against IXP prefixes published by PeeringDB PeeringDB (2017) and Packet Clearing House (PCH) Packet Clearing House (2017). Lastly, we mapped each ASN to its corresponding ORG number using CAIDA's AS-to-ORG mapping dataset Huffaker et al. (2018).

CPs are heterogeneous in handling path measurements. In our mappings, we observed the use of private IP addresses internally by CPs as well as on traceroutes traversing the three connectivity options. We measured the number of observed AS/ORGs (excluding hops utilizing private IP addresses) for inter-cloud, intra-cloud, and cloud-to-LG, and made the following two observations. First, of the three CPs, only AWS used multiple ASNs (i.e. ASes 8987, 14618, and 16509). Second, not surprisingly, we observed a striking difference between how CPs respond to traceroute probes. In particular, we noted that the differences in responses are dependent on the destination network and path type (public vs. private). For example, GCP does not expose any of its routers unless the target address is within another GCP region. Similarly, Azure does not expose its internal routers except for their border routers that are involved in peering with other networks. Finally, we found that AWS heavily relies on private/shared IP addresses for their internal network. These observations serve as motivation for our

characterization of the various multi-cloud connectivity options in § 5.4 and § 5.5 below.

5.3.4 Representation of Results. Distributions in this chapter are presented using letter-value plots Hofmann, Kafadar, and Wickham (2011). Letter-value plots, similar to boxplots, are helpful for summarizing the distribution of data points but offer finer details beyond the quartiles. The median is shown using a dark horizontal line and the $1/2^i$ quantile is encoded using the box width, with the widest boxes surrounding the median representing the quartiles, the 2nd widest boxes corresponding to the octiles, etc. Distributions with low variance centered around a single value appear as a narrow horizontal bar while distributions with diverse values appear as vertical bars.

Throughout this chapter we try to present full distributions of latency when it is illustrative. Furthermore, we compare latency characteristics of different paths using the median and variance measures and specifically refrain from relying on minimum latency as it does not capture the stability and dynamics of this measure across each path.

5.3.5 Ethical and Legal Considerations. This study does not raise any ethical issues. Overall, our goal in this study is to measure and improve multi-cloud connectivity without attributing particular features to any of the utilized third-party providers which might be in violation of their terms of service. Hence, we obfuscate, and wherever possible, omit all information that can be used to identify the colocation and third-party connectivity providers. This information includes names, supported measurement APIs, costs, time and date of measurements, topology information, and any other potential identifiers.

5.4 Characteristics of C2C routes

In this section, we characterize the performance of C2C routes and attribute their characteristics to connectivity and routing.

5.4.1 Latency Characteristics. CPP routes exhibit lower latency than TPP routes and are stable. Figure 23 depicts the distribution of RTT values between different CPs across different connectivity options. The rows (from top to bottom) correspond to AWS, GCP, and Azure as the source CP, respectively. Intra-region (inter-region) measurements are shown in the left (right) columns, and CPP (TPP) paths are depicted in blue (orange). To complement Figure 23, the median RTT values comparing CPP and TPP routes are shown in Figure 24.

From Figures 23 and 24, we see that, surprisingly, CPP routes typically exhibit lower medians of RTT compared to TPP routes, suggesting that CPP routes traverse the CP's optical private backbone. We also observe a median RTT of ∼2ms between AWS and Azure VMs in California which is in accordance with the relative proximity of their datacenters for this region. The GCP VM in California has a median RTT of 13ms to other CPs in California, which can be attributed to the geographical distance between GCP's California datacenter in LA and the Silicon Valley datacenters for AWS and Azure. Similarly, we notice that the VMs in Virginia all exhibit low median RTTs between them. We attribute this behavior to the geographical proximity of the datacenters for these CPs. At the same time, the inter-region latencies within a CP are about 60ms with the exception of Azure which has a higher median of latency of about 67ms. Finally, the measured latencies (and hence the routes) are asymmetric in both directions albeit the median of RTT values in Figure 24 shows latency symmetry (<0.1ms).

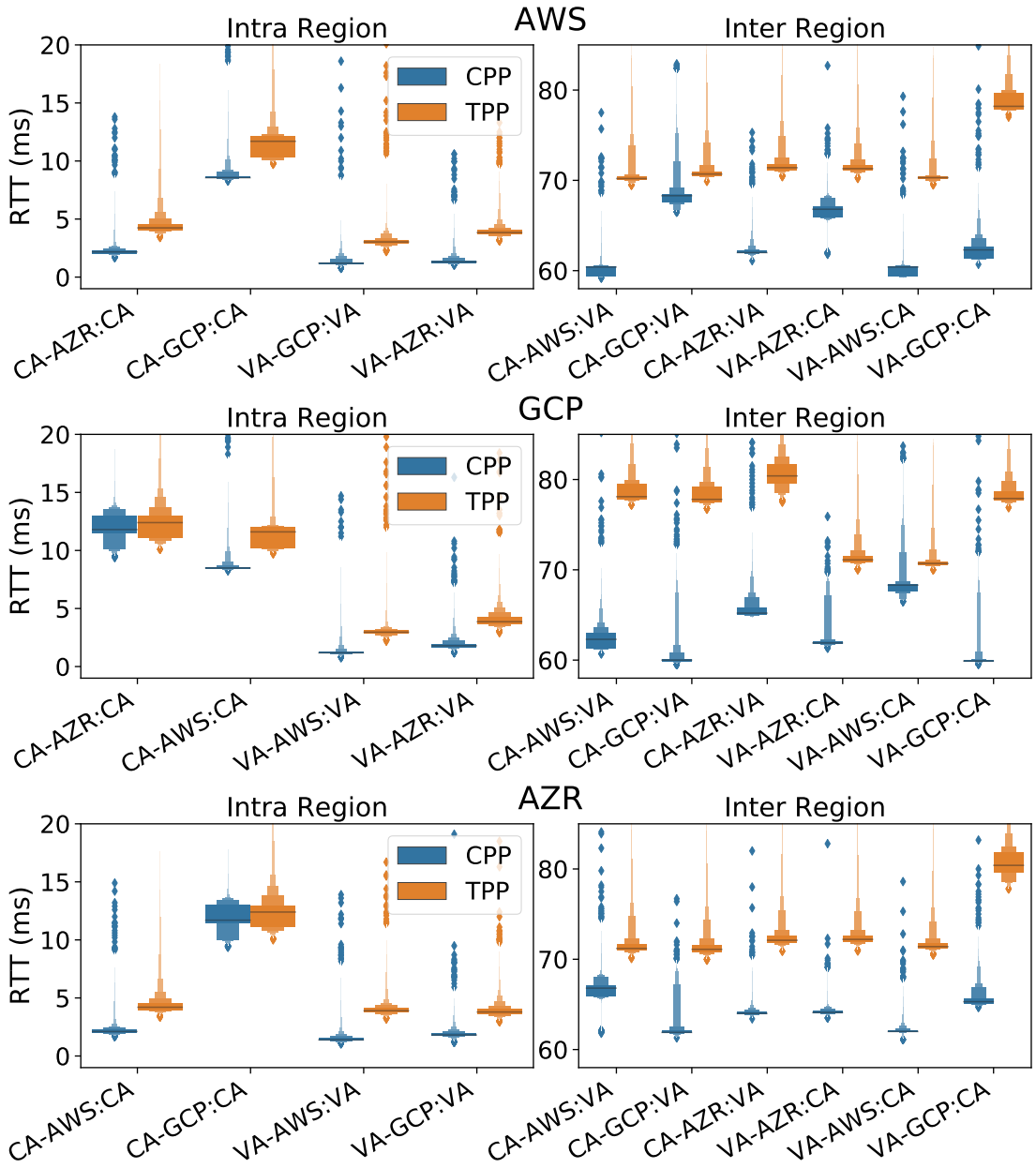


Figure 23. Rows from top to bottom represent the distribution of RTT (using letter-value plots) between AWS, GCP, and Azure's network as the source CP and various CP regions for intra (inter) region paths in left (right) columns. CPP and TPP routes are depicted in blue and orange, respectively. The first two characters of the X axis labels encode the source CP region with the remaining characters depicting the destination CP and region.

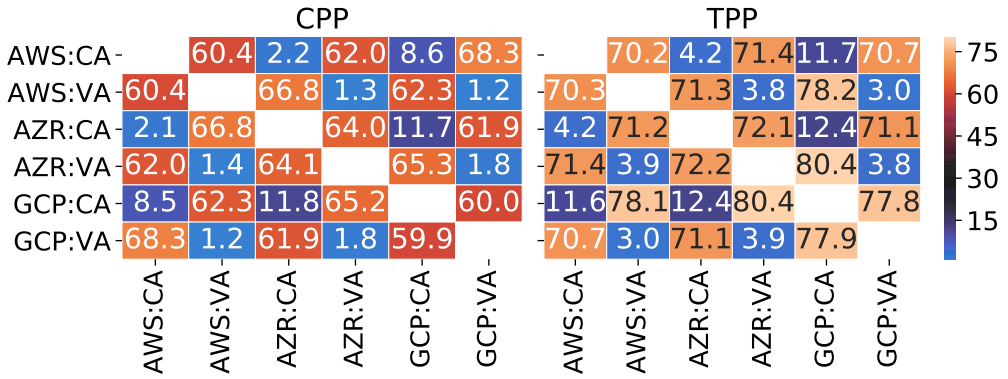


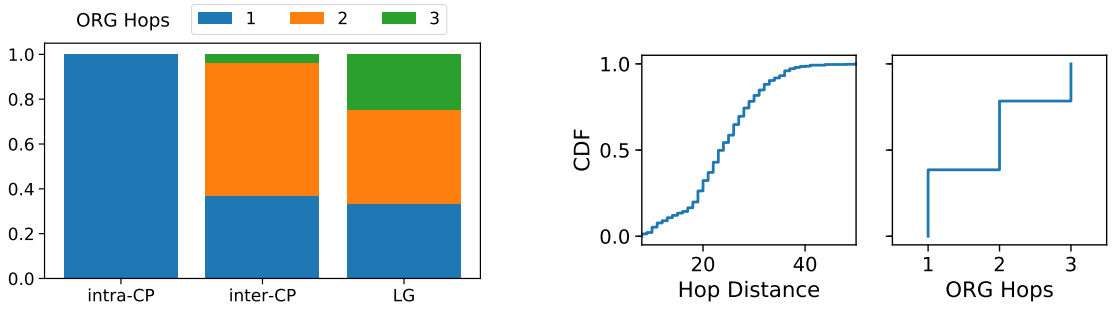
Figure 24. Comparison of median RTT values (in ms) for CPP and TPP routes between different pairs.

Also, the median of the measured latency between our CRs is in line with the published values by third-party connectivity providers, but the high variance of latency indicates that the TPP paths are in general a less reliable connectivity option compared to CPP routes. Lastly, BEP routes for C2LG measurements always have an equal or higher median of latency compared to CPP paths with much higher variability (order of magnitude larger standard deviation). Results are omitted for brevity and to avoid skewed scales in current figures.

5.4.2 Why do CPP routes have better latency than TPP routes?.

CPP routes are short, stable, and private. Figure 25a depicts the distribution of ORG hops for different connectivity options. We observe that intra-cloud paths always have a single ORG, indicating that regardless of the target region, the CP routes traffic internally towards the destination VM. More interestingly, the majority of inter-cloud paths only observe two ORGs corresponding to the source and destination CPs. Only a small fraction (<4%) of paths involves three ORGs, and upon closer examination of the corresponding paths, we find that they traverse IXPs and involve traceroutes that originate from Azure and are destined to Amazon's network in another region. We reiterate that

single ORG inter-CP paths correspond to traceroutes which are originated from GCP's network and does not reveal any internal hops of its network. For the cloud-to-LG paths, we observe a different number of ORGs depending on the source CP as well as the physical location of the target LG. The observations range from only encountering the target LG's ORG to seeing intermediary IXP hops as points of peering. Lastly, we measure the stability of routes at the AS-level and observe that all paths remain consistently stable over time with the exception of routes sourced at Azure California and destined to Amazon Virginia. The latter usually pass through private peerings between the CPs, and only less than 1% of our path measurements go through an intermediary IXP. In short, we did not encounter any transit providers in our measured CPP routes.



(a) (b)

Figure 25. (a) Distribution for number of ORG hops observed on intra-cloud, inter-cloud, and cloud to LG paths. (b) Distribution of IP (AS/ORG) hop lengths for all paths in left (right) plot.

CPs are tightly interconnected with each other in the US. Not observing any transit AS along our measured C2C paths motivated us to measure the prevalence of this phenomenon by launching VM instances within all US regions for our target CP networks. This results in a total of 17 VM instances corresponding to 8, 5, and 4 regions within Azure, GCP, and AWS. We perform

UDP and ICMP *paris-traceroutes* using *scamper* between all VM instances (272 unique pairs) in 10-minute rounds for four days and remove the small fraction ($9 * 10^{-5}$) of traceroutes that encountered a loop along the path. Overall, we observe that ICMP probes are better in revealing intermediate hops as well as reaching the destination VMs. Similar to § 5.3.3, we annotate the hops of the collected traceroutes with their corresponding ASN/ORG and infer the presence of IXP hops along the path. For each path, we measure its IP and AS/ORG hop length and show in Figure 25b the corresponding distributions. C2C paths exhibit a median (0.9 percentile) IP hop length of 22 (33). Similar to our initial C2C path measurements, with respect to AS/ORGhop length, we only observe ORGs corresponding to the three target CPs as well as IXP ASNs for Coresite Any2 and Equinix. All ORG hop paths passing through an IXP correspond to paths which are sourced from Azure and are destined to AWS. The measurements further extend our initial observation regarding the rich connectivity of our three large CPs and their tendency to avoid exchanging traffic through the public Internet.

On the routing models of multi-cloud backbones. By leveraging the AS/ORG paths described in § 5.3, we next identify the peering points between the CPs. Identifying the peering point between two networks from traceroute measurements is a challenging problem and the subject of many recent studies Alexander et al. (2018); Luckie et al. (2016); Marder and Smith (2016). For our study, we utilized the latest version of *bdrmapIT* Alexander et al. (2018) to infer the interconnection segment on the collection of traceroutes that we have gathered. Additionally, we manually inspected the inferred peering segments and, where applicable, validated their correctness using (i) IXP address to tenant ASN mapping and (ii) DNS names such as AMAZON.SJC-96CBE-1A.NTWK.MSN.NET

which is suggestive of peering between AWS and Azure. We find that *bdrmapIT* is unable to identify peering points between GCP and the other CPs since GCP only exposes external IP addresses for paths destined outside of its network, i.e. *bdrmapIT* is unaware of the source CPs network as it does not observe any addresses from that network on the initial set of hops. For these paths, we choose the first hop of the traceroute as the peering point only if it has an ASN equal to the target IP addresses ASN.Using this information, we measure the RTT between the source CP and the border interface to infer the geo-proximity of the peering point from the source CP. Using this heuristic allows us to analyze each CP's inclination to use hot-potato routing.

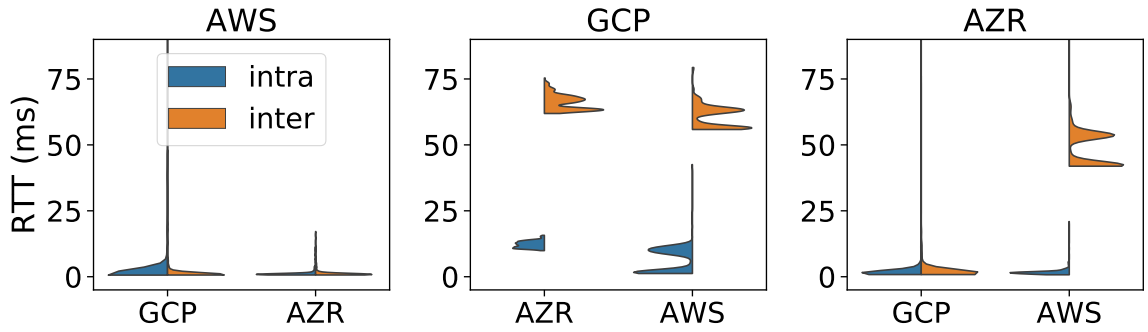


Figure 26. Distribution of RTT between the source CP and the peering hop. From left to right plots represent AWS, GCP, and Azure as the source CP. Each distribution is split based on intra (inter) region values into the left/blue (right/orange) halves, respectively.

Figure 26 shows the distribution of RTT for the peering points between each CP. From left to right, the plots represent AWS, GCP, and Azure as the source CP. Each distribution is split based on intra (inter) region values into the left/blue (right/orange) halves, respectively. We observe that AWS' peering points with other CPs are very close to their networks and therefore, AWS is employing hot-potato routing. For GCP, we find that hot-potato routing is never employed and

traffic is always handed off near the destination region. The bi-modal distribution of RTT values for each destination CP is centered at around 2ms, 12ms, 58ms, and 65ms corresponding to the intra-region latency for VA and CA, inter-region latency to GCP, and inter-region latency to other CPs, respectively. Finally, Azure exhibits mixed routing behavior. Specifically, Azure's routing behavior depends on the target network – Azure employs hot-potato routing for GCP, its Virginia-California traffic destined to AWS is handed off in Los Angeles, and for inter-region paths from California to AWS Virginia, the traffic is usually (99%) handed off in Dallas TX and for the remainder is being exchanged through Digital Realty Atlanta's IXP.

From these observations, the routing behavior for each path can be modeled with a simple threshold-based method. More concretely, for each path i with an end-to-end latency of l_{ei} and a border latency of l_{bi}, we can infer if source CP employs hot-potato routing if $l_{bi} < \frac{1}{10} l_{ei}$. Otherwise, the source CP employs cold-potato routing (i.e. $l_{bi} > \frac{9}{10} l_{ei}$). The fractions (i.e. $\frac{1}{10}$ and $\frac{9}{10}$) are not prescriptive and are derived based on the latency distributions depicted in Figure 26.

5.4.3 Throughput Characteristics. CPP routes exhibit higher and more stable throughput than TPP routes. Figure 27 depicts the distribution of throughput values between different CPs using different connectivity options. While intra-region measurements tend to have a similar median and variance of throughput, we observe that with respect to inter-region measurements, TPPs exhibit a lower median throughput with higher variance. Degradation of throughput seems to be directly correlated with higher RTT values as shown in Figure 23. Using our latency measurements, we also approximate loss-rate to be 10^{-3} and 10^{-4} for TPP and CPP routes, respectively. Using the formula of

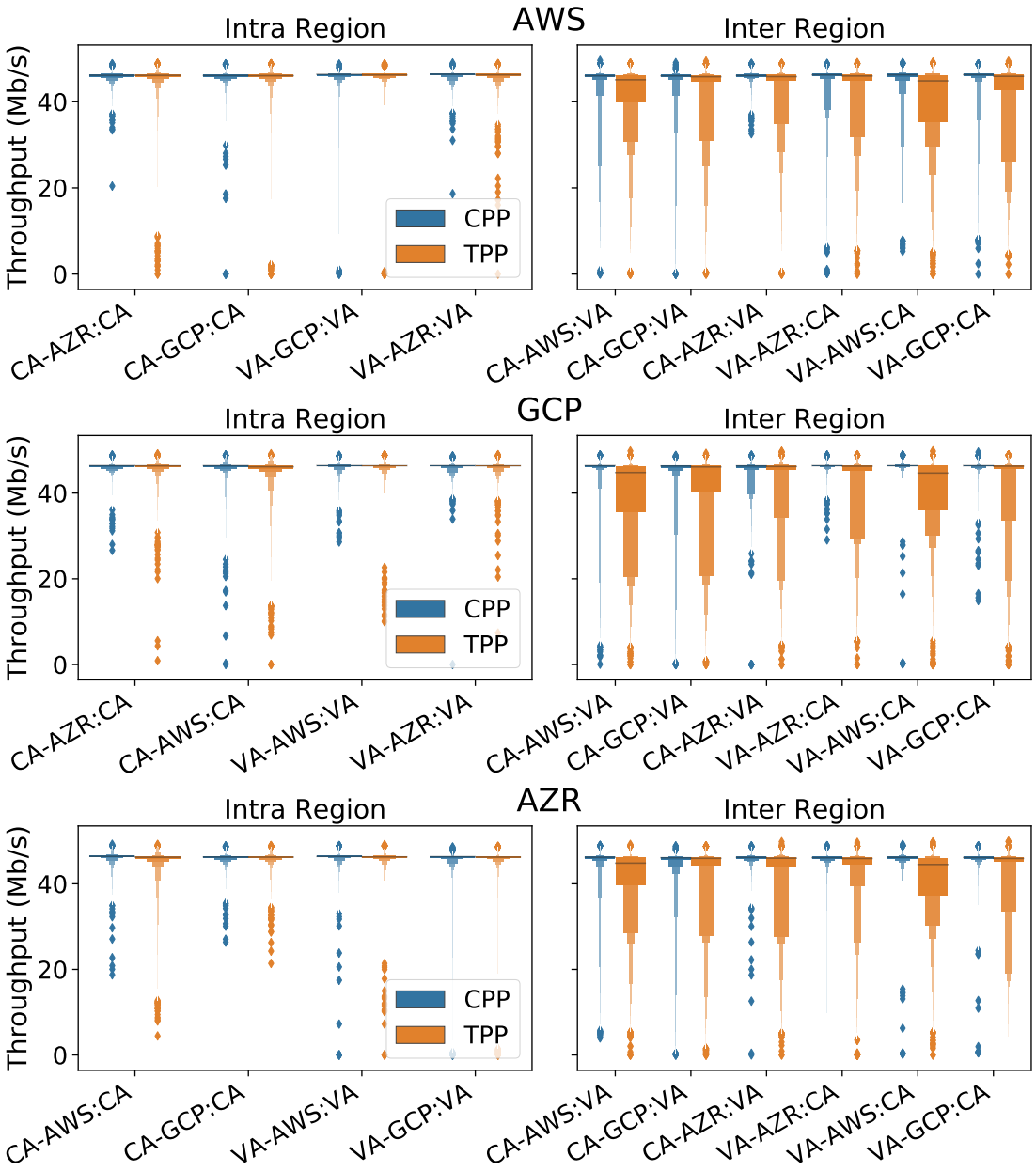


Figure 27. Rows from top to bottom in the letter-value plots represent the distribution of throughput between AWS', GCP's, and Azure's network as the source CP and various CP regions for intra- (inter-) region paths in left (right) columns. CPP and TPP routes are depicted in blue and orange respectively.

Mathis et al. Mathis et al. (1997) to approximate TCP throughput[4], we can obtain

[4]We do not have access to parameters such as TCP timeout delay and number of acknowledged packets by each ACK to use more elaborate TCP models (e.g. Padhye, Firoiu, Towsley, and Kurose (1998)).

an upper bound for throughput for our measured loss-rate and latency values. Figure 28 shows the upper bound of throughput for an MSS of 1460 bytes and several modes of latency and loss-rate. For example, the upper bound of TCP throughput for a 70ms latency and loss-rate of 10^{-3} (corresponding to the average measured values for TPP routes between two coasts) is about 53Mb/s. While this value is higher than our interface/link bandwidth cap of 50Mb/s, bursts of packet loss or transient increases in latency could easily lead to sub-optimal TCP throughput for TPP routes.

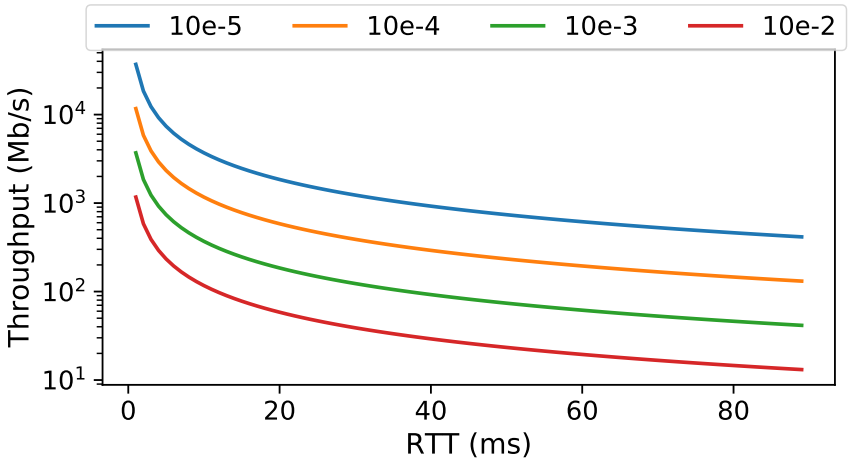


Figure 28. Upper bound for TCP throughput using the formula of Mathis et al. Mathis et al. (1997) with an MSS of 1460 bytes and various latency (X axis) and loss-rates (log-scale Y axis) values.

5.4.4 Why do CPP routes have better throughput than TPP routes?. TPPs have higher loss-rates than CPPs.

Our initial methodology for measuring loss-rate relied on our low-rate *ping* probes (outlined in § 5.3.3). While this form of probing can produce a reliable estimate of average loss-rate over a long period of time Tariq, Dhamdhere, Dovrolis, and Ammar (2005), it doesn't capture the dynamics of packet loss at finer resolutions. We thus modified our

probing methodology to incorporate an additional *iperf3* measurement using UDP probes between all CP instances. Each measurement is performed for 5 seconds and packets are sent at a 50Mb/s rate.[5] We measure the number of transmitted and lost packets during each second and also count the number of packets that were delivered out of order at the receiver. We perform these loss-rate measurements for a full week. Based on this new set of measurements, we estimate the overall loss-rate to be $5 * 10^{-3}$ and 10^{-2} for CPP and TPP paths, respectively. Moreover, we experience 0 packet loss in 76% (37%) of our sampling periods for CPP (TPP) routes, indicating that losses for CPP routes tend to be more bursty than for TPP routes. The bursty nature of packet losses for CPP routes could be detrimental to real-time applications which can *only* tolerate certain levels of loss and should be factored in by the client. The receivers did not observe any out-of-order packets during our measurement period.

Figure 29 shows the distribution of loss rate for various paths. The rows (from top to bottom) correspond to AWS, GCP, and Azure as the source CP, respectively. Intra-region (inter-region) measurements are shown in the left (right) columns, and CPP (TPP) paths are depicted in blue (orange). We observe consistently higher loss-rates for TPP routes compared to their CPP counterparts and lower loss-rates for intra-CP routes in Virginia compared to California. Moreover, paths destined to VMs in the California region show higher loss-rates regardless of where the traffic has been sourced from, with asymmetrically lower loss-rate on the reverse path indicating the presence of congested ingress points for CPs within the California region. We also notice extremely low loss-rates for intra-CP (except Azure) CPP routes between the US east and west coasts and for

[5]In an ideal setting, we should not experience any packet losses as we are limiting our probing rate at the source.

inter-CP CPP routes between the two coasts for certain CP pairs (e.g. AWS CA to GCP VA or Azure CA to AWS VA).

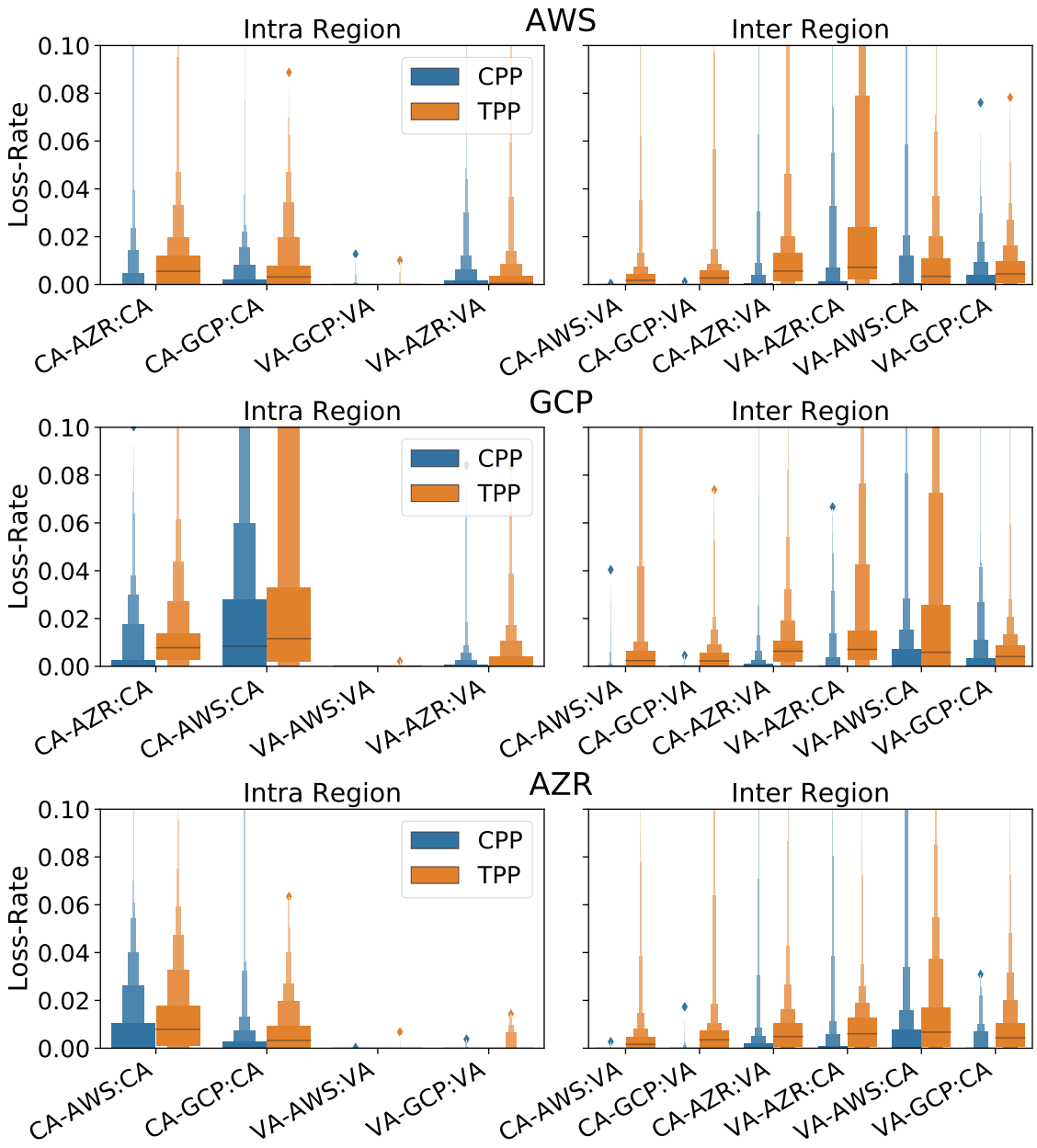


Figure 29. Rows from top to bottom in the letter-value plots represent the distribution of loss-rate between AWS, GCP, and Azure as the source CP and various CP regions for intra- (inter-) region paths in left (right) columns. CPP and TPP routes are depicted using blue and orange respectively.

5.4.5 Summary. To summarize, our measurements for characterizing C2C routes reveal the following important insights:

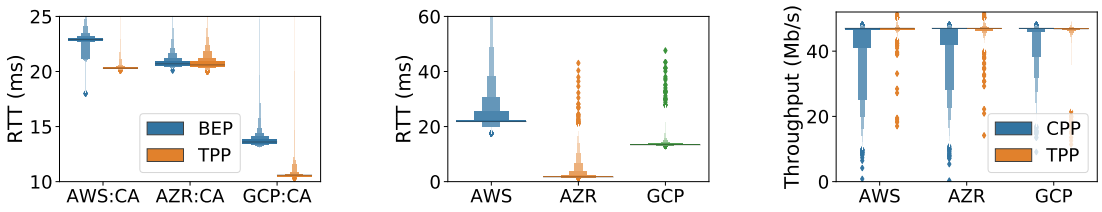
- CPP routes are better than TPP routes in terms of latency as well as throughput. This finding begs the question: *Given the sub-optimal performance of TPP routes and their cost implications, why should an enterprise seek connectivity from third-party providers when deciding on its multi-cloud strategy?*
- The better performance of CPP routes as compared to their TPP counterparts can be attributed to two factors: (a) the CPs' rich (private) connectivity in different regions with other CPs (traffic is by-passing the BEP Internet altogether) and (b) more stable and better provisioned CPP (private) backbones.

5.5 Characteristics of E2C routes

In this section, we turn our attention to E2C routes, characterize their performance and attribute the observations to connectivity and routing.

5.5.1 Latency Characteristics. TPP routes offer better latency than BEP routes. Figure 30a shows the distribution of latency for our measured E2C paths. We observe that TPP routes consistently outperform their BEP counterparts by having a lower baseline of latency and also exhibiting less variation. We observe a median latency of 11ms, 20ms, and 21ms for TPP routes towards GCP, AWS, and Azure VM instances in California, respectively. We also observe symmetric distributions on the reverse path but omit the results for brevity.

5.5.2 Why do TPP routes offer better latency than BEP routes?. In the case of our E2C paths, we always observe direct peerings between the upstream provider (e.g. Cox Communications (AS22773)) and the CP network.



(a) (b) (c)

Figure 30. (a) Distribution of latency for E2C paths between our server in AZ and CP instances in California through TPP and BEP routes. Outliers on the Y-axis have been deliberately cut-off to increase the readability of distributions. (b) Distribution of RTT on the inferred peering hop for E2C paths sourced from CP instances in California. (c) Distribution of throughput for E2C paths between our server in AZ and CP instances in California through TPP and BEP routes.

Relying on *bdrmapIT* to infer the peering points from the traceroutes associated with our E2C paths, we measure the latency on the peering hop. Figure 30b shows the distribution of the latency for the peering hop for E2C paths originated from the CPs' instances in CA towards our enterprise server in AZ. While the routing policies of GCP and Azure for E2C paths are similar to our observations for C2C paths, Amazon seems to hand-off traffic near the destination which is unlike their hot-potato tendencies for C2C paths. We hypothesize that this change in AWS' policy is to minimize the operational costs via their Transit Gateway service Amazon (2019b). In addition, observing an equal or lower minimum latency for TPP routes as compared to BEP routes suggests that TPP routes are shorter than BEP paths[6]. We also find (not shown here) that the average loss rate on TPP routes is $6 * 10^{-4}$ which is an order of magnitude lower than the loss rate experienced on BEP routes ($1.6 * 10^{-3}$).

5.5.3 Throughput Characteristics. TPP offers consistent throughput for E2C paths. Figure 30c depicts the distribution of throughput

[6]In the absence of information regarding the physical fiber paths, we rely on latency as a proxy measure of path length.

for E2C paths between our server in AZ and CP instances in CA via TPP and BEP routes, respectively. While we observe very consistent throughput values near the purchased link capacity for TPP paths, BEP paths exhibit higher variability which is expected given the best effort nature of public Internet paths.

5.5.4 Summary. In summary, our measurements for characterizing E2C routes support the following observations:

- TPP routes exhibit better latency and throughput characteristics when compared with BEP routes.
- The key reasons for the better performance of TPP routes as compared to their BEP counterparts include shorter (e.g. no transit providers) and better performant (e.g. lower loss rate) paths.
- For an enterprise deciding on a suitable multi-cloud strategy, CPP routes are better *only* when enterprises are closer to the CPs' native locations. Given that TPPs are present at many geographic locations where the CPs are not native, third-party providers offer better connectivity options compared to relying on the public Internet (i.e. using BEP routes).

5.6 Discussion and Future Work

In this section, we discuss the limitations of our study and open issues. We also discuss ongoing and future work.

Representativeness. While the measurement setup depicted in Figure 22 represents a realistic enterprise network employing a multi-cloud strategy, it is not the *only* representative setting. We note that there are a number of other multi-cloud connectivity scenarios (e.g. distinct CPs in different continents, different third-party providers in different countries, etc.), which we do not discuss in this study. For example, what are the inter-cloud connectivity and routing

characteristics between intercontinental VMs e.g. in USA and EU? Unfortunately, the costs associated with establishing TPP paths prevent an exhaustive exploration of multi-cloud connectivity in general and TPP connectivity in particular.

Additional Cloud and Third-party Providers. Our study focuses on multi-cloud connectivity options between three major CPs (i.e. AWS, Azure, and GCP) as they collectively have a significant market share. We plan to consider additional cloud providers (e.g. Alibaba, IBM Softlayer, Oracle, etc.) as part of future work.

Similar to the availability of other CPs, TPP connectivity between CPs are offered via new services by a number of third-party connectivity providers Amazon (2018c); Google (2018b); Microsoft (2018c). Exploring the TPP connectivity provided by the ecosystem and economics of these different third-party providers is an open problem. In addition, there has been no attempt to date to compare their characteristics in terms of geography, routing, and performance, and we intend to explore this aspect as part of future work.

Longitudinal Analysis & Invariants. Despite the fact that we conduct our measurements for about a month in the Spring of 2019 (as mentioned in § 5.3.5), we note that our study is a short-term characterization of multi-cloud connectivity options. Identifying the invariants in this context requires a longitudinal analysis of measurements which is the focus of our ongoing work.

Impact of Connectivity Options on Cloud-hosted Applications. Modern cloud applications pose a wide variety of latency and throughput requirements. For example, key-value stores are latency sensitive Tokusashi, Matsutani, and Zilberman (2018), whereas applications like streaming and geo-distributed analytics require low latency as well as high throughput Lai,

Chowdhury, and Madhyastha (2018). In the face of such diverse requirements, what is critically lacking is a systematic benchmarking of the impact of performance tradeoffs between the BEP, CPP and TPP routes on the cloud-hosted applications (e.g. key-value stores, streaming, etc.) While tackling WAN heterogeneity is the focus of a recent effort Jonathan, Chandra, and Weissman (2018), dealing with multi-cloud connectivity options and their impacts on applications is an open problem.

Connectivity and Routing Implications. In terms of routing and connectivity, our study has two implications. First, while it is known that the CPs are contributing to the ongoing “flattening" of the Internet Dhamdhere and Dovrolis (2010); Gill, Arlitt, Li, and Mahanti (2008); Labovitz, Iekel-Johnson, McPherson, Oberheide, and Jahanian (2010), our findings underscore the fact that the third-party private connectivity providers act as a catalyst to the ongoing flattening of the Internet. In addition, our study offers additional insights into the ongoing “cloudification" of the Internet in terms of where and why cloud traffic bypasses the BEP transits. Our study also implies that compared to the public Internet, CPP backbones are better performant, more stable, and more secure (invisible and isolated from the BEP transits), making them first-class citizens for future Internet connectivity. In light of these two implications, our study also warrants revisiting existing efforts from the multi-cloud perspective. In particular, we plan to pursue issues such as failure detection and characterization for multi-cloud services (e.g. Zhang, Zhang, Pai, Peterson, and Wang (2004)) and multi-cloud reliability (e.g. Quan, Heidemann, and Pradkin (2013)). Other open problems concern inferring inter-CP congestion (e.g. Dhamdhere et al. (2018)) and examining

the economics of multi-cloud strategies (e.g. Zarchy, Dhamdhere, Dovrolis, and Schapira (2018)).

5.7 Summary

Enterprises are connecting to multiple CPs at an unprecedented pace and multi-cloud strategies are here to stay. Due to this development, in addition to best-effort public (BEP) transit provider-based connectivity, two additional connectivity options are available in today's Internet: third-party private (TPP) connectivity and cloud-provider private (CPP) connectivity.

In this work, we perform a first-of-its-kind measurement study to understand the tradeoffs between three popular multi-cloud connectivity options (CPP vs. TPP vs. BEP). Based on our cloud-centric measurements, we find that CPP routes are better than TPP routes in terms of latency as well as throughput. The better performance of CPPs can be attributed to (a) CPs' rich connectivity in different regions with other CPs (by-passing the BEP Internet altogether) and (b) CPs' stable and well-designed private backbones. In addition, we find that TPP routes exhibit better latency and throughput characteristics when compared with BEP routes. The key reasons include shorter paths and lower loss rates compared to the BEP transits. Although limited in scale, our work highlights the need for more transparency and access to open measurement platforms by all the entities involved in interconnecting enterprises with multiple clouds.

CHAPTER VI

OPTIMAL CLOUD OVERLAYS

Motivated by the observations in Chapter V on the diversity of performance characteristics of various cloud connectivity paths, in this chapter, we design an extensible measurement platform for cloud environments. Furthermore, we create a decision support framework that facilitates enterprises in creating optimal multi-cloud deployments.

The content in this chapter is the result of a collaboration between Bahador Yeganeh with Ramakrishnan Durairajan, Reza Rejaie, and Walter Willinger. Bahador Yeganeh is the primary author of this work and responsible for desiging all systems, conducting measurements and producing the presented analyses.

6.1 Introduction

Modern enterprises are adopting multi-cloud strategies[1] at a rapid pace. Among the benefits of pursuing such strategies are competitive pricing, vendor lockout, global reach, and requirements for data sovereignty. According to a recent industry report, more than 85% of enterprises have already adopted multi-cloud strategies Krishna et al. (2018).

Despite this existing market push for multi-cloud strategies, we posit that there is a technology pull: *seamlessly connecting resources across disparate, already-competitive cloud providers (CPs) in a performance- and cost-aware manner is an open problem.* This problem is further complicated by two keys issues. First, prior research on overlays has focused either on the public Internet-based Andersen, Balakrishnan, Kaashoek, and Morris (2001) or on CP paths in isolation Costa, Migliavacca, Pietzuch, and Wolf (2012); Haq, Raja, and Dogar (2017); Lai et al.

[1]This is different from hybrid cloud computing, where a direct connection exists between a public cloud and private on-premises enterprise server(s).

(2018). Second, because CP backbones are private and are invisible to traditional measurement techniques, we lack a basic understanding of their performance, path, and traffic-cost characteristics.

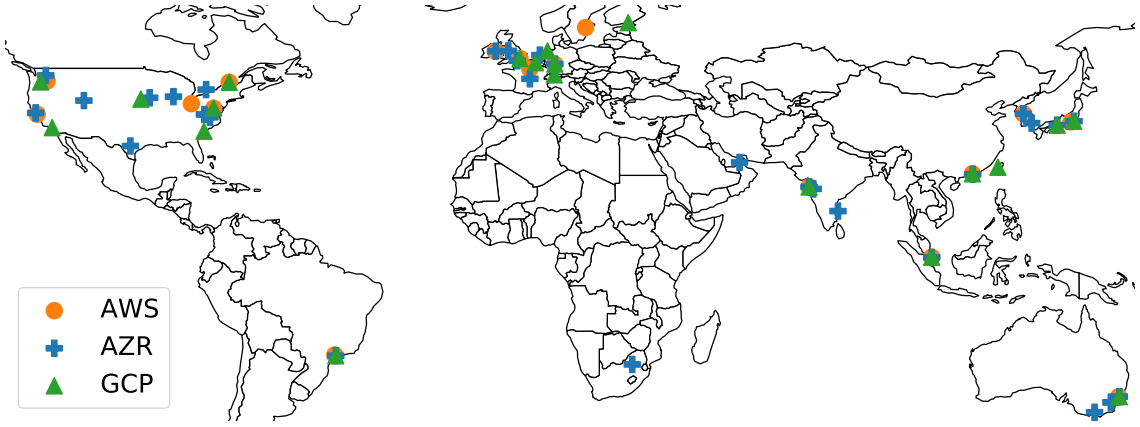


Figure 31. Global regions for AWS, Azure, and GCP.

To examine the benefits of multi-cloud overlays, we perform a third-party, cloud-centric measurement study[2] to understand the performance, path, and traffic-cost characteristics of three major global-scale private cloud backbones (i.e., AWS, Azure and GCP). Our measurements were ran across 6 continents and 23 countries for 2 weeks (see Figure 31). Our measurements reveal a number of key insights. First, the cloud backbones (a) are optimal (i.e., 2x reduction in latency inflation ratio, which is defined as the ratio between line-of-sight and latency-based speed-of-light distances, w.r.t. public Internet), (b) lack path and delay asymmetry, and (c) are tightly interconnected with other CPs. Second, multi-cloud paths exhibit higher latency reductions than single cloud paths; e.g., 67% of all paths, 54% of all intra-CP paths, and 74% of all inter-CP paths experience an improvement in their latencies. Third, although traffic costs vary from location to location and

[2]Code and datasets used in this study will be openly available to the community upon publication.

across CPs, the costs are not prohibitively high. Based on these insights, we argue that enterprises and cloud users can indeed benefit from future efforts aimed at constructing high-performance overlay networks atop multi-cloud underlays in a performance- and cost-aware manner.

While our initial findings suggest that multi-cloud overlays are indeed beneficial for enterprises, establishing overlay-based connectivity to route enterprise traffic in a cost- and performance-aware manner among islands of disparate CP resources is an open and challenging problem. For one, the problem is complicated by the lack of continuous multi-cloud measurements and vendor-agnostic APIs.

To tackle these challenges, the main goal of this chapter is to create a service to establish and manage overlays on top of multi-cloud underlays. The starting point of our approach to create a cloud-centric measurement and management service called *Tondbaz* that continuously monitors the inter- and intra-CP links. At the core of *Tondbaz* are vendor-agnostic APIs to connect the disparate island of CP resources. With the measurement service and APIs in place, *Tondbaz* constructs a directed graph consisting of nodes that represent VM instances, given two locations (e.g., cities) as input by a cloud user. Edges in the graph will be annotated with latencies and traffic-cost values from the measurement service.

This study makes the following contributions:

- We propose and design an extensible system called *Tondbaz* to facilitate the measurement of multi-CP network paths.
- We design a decision-support framework for constructing optimal cloud overlay paths using insights gleaned using *Tondbaz*.

– We demonstrate the cost and performance benefits of utilizing a decision-support framework by integrating it into the *snitch* mechanism of Cassandra, a distributed key-value store.

The remainder of this chapter is organized by, first presenting the design objectives of our measurement platform and provide formal definitions for our optimization framework in §6.2. In §6.3 we utilize our measurement platform to measure the path characteristics of the top 3 CPs on a global scale and apply our optimization framework to obtain optimal paths between all pairs of CP regions. Next, we demonstrate the applicability of our overlays for a handful of paths and discuss the operational trade-offs of overlays in §6.4.2. Lastly, we conclude this chapter by summarizing our findings in §6.5.

6.2 *Tondbaz* Design

In this section, we will describe the *Tondbaz*'s components and their corresponding design principles and objectives. At a high-level *Tondbaz* consists of 2 main components namely, (i) a measurement platform for conducting cloud to cloud measurements 6.2.1 and (ii) a decision support framework for obtaining optimal cloud paths based on a set of constraints 6.2.3.

6.2.1 Measurement Platform. The measurement platform is designed with low resource overhead and extensibility as objectives in mind. The measurement platform consists of the following 3 main components:

– an *agent* for conducting/gathering multi-cloud performance measurements

– a centralized data-store for collecting and archiving the measurement results from each *agent*

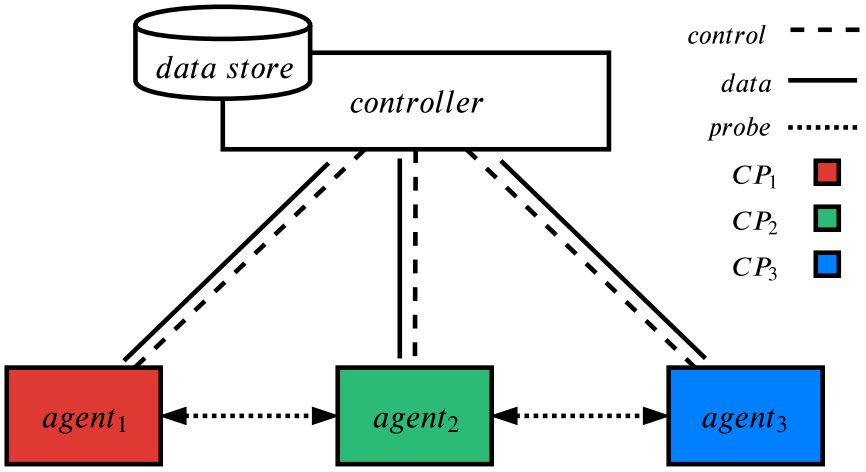


Figure 32. Overview of components for the measurement system including the centralized controller, measurement *agents*, and data-store.

– a centralized controller/scheduler that configures each measurement *agent* and schedules measurement tasks

Figure 32 shows a high-level overview of the components for the measurement platform as well as how they communicate with each other. The *agents* communicate with the centralized *controller* in a client-server model over a control channel. Furthermore, the *agents* store the result of running the measurements in a *data-store*. The *data-store* and *controller* by design are decoupled from each other although they can reside on the same node.

6.2.1.1 Measurement Agent. The measurement *agent* is designed with multiple objectives in mind namely, (i) ease of deployment, (ii) low resource overhead, (iii) and extensibility of measurements. In the following, we describe how each of these design objectives are achieved within our measurement *agents*.

Ease of Deployment: The measurement subsystem is designed to be installed as a daemon on the host system with minimal dependencies (except for a *python* distribution) using a simple shell script. The only required parameter for

installation is the address for the *controller* that the *agent* will be communicating with. Upon installation, the *agent* would announce itself to the *controller* on a predefined channel. After registration with the *controller* configuration of the *agent* including (but not limited to) target addresses, output destination, execution of measurement tasks should all happen through a configuration channel and therefore can be managed from a centralized location. We rely on the MQTT protocol OASIS (2019) for communication between *agents* and the centralized *controller*.

Low Resource Overhead: A barebone *agent* is simply a daemon listening for incoming commands from the centralized *controller* on its control channel. Using this minimal design the *agent* is implemented in less than 1k lines of python code with a single dependency on the Eclipse Paho MQTT library Eclipse (2019). The *agent* uses 10MB of memory on runtime and requires less than 100KB of memory to maintain the state for ongoing measurement.

Extensibility of Measurements: The *agents* should support a wide range of measurements including standard network measurement tools such as *ping, traceroute, iperf* as well as any custom executables. Each measurement tool should be implemented as a container image. In addition to the container image, the developer should implement a Python class that inherits from a standard interface depicting how the *agent* can communicate with the measurement tool. Measurement results should be serialized into a predefined JSON schema prior to being stored on the *data-store*.

6.2.1.2 Centralized Controller.

The coordinator awaits incoming connections from *agents* that announce their presence and register themselves with the coordinator (*anc*). After the initial registration the coordinator can schedule and conduct measurements on the *agent* if needed. The coordinator would maintain

a control channel with each *agent* which is used for (i) monitoring the health of each *agent* through heartbeat messages (*hbt*), (ii) sending configuration parameters (*cfg*), (iii) scheduling and issuing measurement commands (*run* and *fin*), and (iv) monitoring/reporting the status of ongoing measurements (*sta*).

6.2.2 Data Collector. *Tondbaz agents* can store the results of each measurement locally for later aggregation in a centralized data-store. Additionally, each *agent* can stream the results of each measurement back to the centralized data-store. Each measurement result is presented as a JSON object containing generic fields (start time, end time, measurement id, agent address) in addition to a JSON serialized representation of the measurement output provided by the commands plugin for *Tondbaz agent*. We rely on MongoDB for our centralized data collector given that our data has a NoSQL JSON schema.

6.2.3 Optimization Framework. In addition to the measurement platform, we have designed an optimization framework that can identify cloud overlay paths that optimize a network performance metric while satisfying certain constraints specified by the user. The optimization framework relies on the stream of measurements reported by all *agents* within the *data-store* and using them would create an internal model of the network using a directed graph G where nodes represent *agent* instances and edges depict the network path between each instance.

$$\begin{gathered} G = (V, E) \\ V = \{v_1, v_2, ..., v_N\} \\ E = \{e_{ij} = (v_i, v_j) |\ \forall v_i, v_j \in V\ (v_i, v_j) \neq (v_j, v_i)\} \end{gathered} \tag{6.1}$$

Measurement results pertaining to the network path are added as edge attributes. Additionally, the optimization framework relies on an internal cost model that calculates the cost of transmitting traffic over each path based on

the policies that each CP advertises on their websites Amazon (2019c); Google (2019); Microsoft (2019). The details of each CP's pricing policy differs from one to another but at a high-level is governed by 4 common rules namely, (i) CPs only charge for egress traffic from a compute instance, (ii) customers are charged based on the volume of exchanged traffic (ii) traffic remaining within a CPs network has a lower charge rate, (iii) each source/destination region (or a combination of both) has a specific charging rate. While the measurement platform is designed to be extensible and supports a wide variety of measurement tools, the optimization framework only utilizes latency and cost measurements. Extensions to the framework to support additional network metrics is part of our future work.

The optimization framework requires the user to specify a series of constraints namely, (i) set of target regions where the user needs to have a deployment (R), (ii) set of regions that should be avoided when constructing an optimal path (A), (iii) a set of region pairs that would be communicating with each other through the overlay (T), and (iv) an overall budget for traffic cost (B). Equation (6.2) formally defines the aforementioned constraints. This formulation of the optimization problem can be mapped to the Steiner tree graph problem Hwang and Richards (1992) which is known to be NP-complete.

$$\begin{gathered} A \subset V \\ V' = V - A \\ R \subset V' \\ T_{ij} = (v_i, v_j);\ \forall v_i, v_j \in R \end{gathered} \tag{6.2}$$

We approximate the solution (if any) to this optimization problem by (i) creating an induced graph by removing all regions in A from its internal directed

graph G (Equation (6.3))

$$\begin{gathered} G' = (V', E') \\ V' = V - A \ , \ E' = (v_i, v_j) \ \forall v_i, v_j \in V' \end{gathered} \tag{6.3}$$

(ii) performing a breadth first search (BFS) to obtain all paths (P) between each pair of regions within T that have an overall cost (C function) within the budget B and do not have an inflated end-to-end latency compared to the default path (l_{ij}) (Equations (6.4) and (6.5))

$$\begin{gathered} P_{ij} = \{p_{xij} |\ p_{xij} = (v_{x1}, ..., v_{xn}), \\ \forall 1 \leq k < n \ (v_{xk}, v_{xk+1}) \in E' \ and \ v_{x1} = v_i, \ v_{xn} = v_j, \\ 1 \leq x \leq \lfloor (|V| - 2)! e \rfloor \} \end{gathered} \tag{6.4}$$

$$\begin{gathered} P'_{ij} = \{p_{xij} |\ C(p_{xij}) \leq B, \ L(p_{xij}) \leq l_{ij}\} \\ C(p_{xij}) = \sum c_{wz}; \ \forall e_{wz} \in p_{xij} \\ L(p_{xij}) = \sum l_{wz}; \ \forall e_{wz} \in p_{xij} \end{gathered} \tag{6.5}$$

and (iii) selecting the overlay that has the overall greatest reduction in latency among all possible sets of overlays (Equation (6.6)).

$$\begin{gathered} O = \{p_{ij} |\ \forall p_{ij} \in P'_{ij} \ and \ \forall i, j \ e_{ij} \in T\} \\ L(O) = \sum l_{ij} - L(p_{ij}) \ ; \ \forall p_{ij} \in O \\ OPT = O_x; \ x = argmin(L(O_x)) \end{gathered} \tag{6.6}$$

The time complexity of this approach is equal to performing a BFS ($O(|V'| + |E'|)$) for each pair of nodes in T in addition to selecting the set of paths which result in the most amount of overall latency reduction. The latter step has a time complexity of $O((|P'|!)^{|T|})$, where $|P'| = \lfloor (|V'| - 2)! e \rfloor$. While the high complexity of the second step might seem intractable, our BFS algorithm

would backtrack whenever it encounters a path that exceeds our total budget B or has an end-to-end latency greater than the default path l_{ij}. Additionally, based on our empirical evaluation we observe that each relay point can add about 1ms of forwarding latency and therefore our search would backtrack from paths that yield less than 1ms of latency improvement per relay hop. Through our analysis, we observed that on average 31% number of paths that do not exceed the default end-to-end latency with an unlimited budget effectively making our solution tractable.

6.3 A Case for Multi-cloud Overlays

In this section, we demonstrate the use of *Tondbaz* to conduct path and latency measurements in a multi-cloud setting (§ 6.3.1), followed by the optimality of single CP paths (§ 6.3.2) and motivating performance gains of multi-cloud paths (§ 6.3.3). Next, we present the challenge of inferring traffic cost profiles which hinders the realization of multi-cloud overlays (§ 6.3.4). Lastly, we investigate the possibility of utilizing IXP points for the creation of further optimal overlays in § 6.3.6.

6.3.1 Measurement Setting & Data Collection. We target the top 3 CPs namely, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). We create small VM instances within all global regions of these CPs resulting in a total of 68 regions (17, 31, and 20 for AWS, Azure, and GCP respectively). regions are dedicated to government agencies and are not available to the public. Furthermore, we were not able to allocate VMs in 5 Azure regions[3]. Through our private correspondence with the support team, we learned that those regions either are mainly designed for storage redundancy of nearby regions or did not have free resources available at the time of this study.

[3]Central India, Canada East, France South, South Africa West, and Australia Central

Additionally, we identify the datacenter's geo-location for each CP. Although CPs are secretive with respect to the location of their datacenters, various sources do point to their exact or approximate location Build Azure (2019); Burrington (2016); Google (2019); Miller (2015); Plaven (2017); WikiLeaks (2018); Williams (2016) and in the absence of any online information we resort to the nearest metro area that the CP advertises.

We conduct pairwise latency and path measurements between all VM instances in 10 minute rounds for the duration of 2 weeks in October of 2019 resulting in about 20k latency and path samples between each pair of VM. Each round of measurement consists of 5 latency probes and 2 (UDP, and TCP) *paris-traceroute* path measurements. The resultant traceroute hops from our path measurements are annotated with their corresponding ASN using BGP feeds of Routeviews University of Oregon (2018) and RIPE RIPE (2018) collectors aggregated by BGPStream Orsini, King, Giordano, Giotsas, and Dainotti (2016). Furthermore, we map each hop to its owner ORG by relying on CAIDA's AS-to-ORG dataset Huffaker et al. (2018). Lastly, the existence of IXP hops along the path is checked by matching hop addresses against the set of IXP prefixes published by PeeringDB PeeringDB (2017), Packet Clearing House (PCH) Packet Clearing House (2017), and Hurricane Electric (HE) using CAIDA's aggregate IXP dataset CAIDA (2018).

6.3.2 Are Cloud Backbones Optimal?.

6.3.2.1 Path Characteristics of CP Backbones. As mentioned above, we measure the AS and ORG path for all of the collected traceroutes. In all our measurements, we observe multiple ASes for AWS *only* (AS14618 and AS16509). Hence, without the loss of generality, from this point onward we only

present statistics using the ORG measure. We measure the ORG-hop length for all unique paths and find that for 97.86% of our measurements, we only observe 2 ORGs (i.e. the source and destination CP networks). Out of the remaining paths, we observe that 2.12%, and 0.02% have 3, and 4 ORG hops respectively. These observations indicate two key results. First, all intra-CP measurements (and, hence, traffic) remain *almost always* within the CPs' backbones. Second, the CP networks are tightly interconnected with each other and establish private peerings between each other on a global scale. Surprised by these findings, we take a closer look at the 2.14% of paths which include other networks along their path. About 76% of these paths have a single IXP hop between the source and destination CPs. That is, the CPs are peering directly with each other over an IXP fabric. For the remaining 24% of paths, we observe 2 prominent patterns: (i) paths sourced from AWS in Seoul and Singapore as well as various GCP regions that are destined to Azure in UAE; and (ii) paths sourced from various AWS regions in Europe and destined to Azure in Busan, Korea.

Main findings: *All intra-CP and the majority of inter-CP traffic remains within the CPs' network and is transmitted between the CPs' networks over private and public peerings. CP's backbones are tightly interconnected and can be leveraged for creating a global multi-cloud overlay.*

6.3.2.2 Performance Characteristics of CP Backbones. Using the physical location of datacenters for each CP, we measure the geo-distance between each pair of regions within a CP's network using the Haversine distance Robusto (1957) and approximate the optimal latency using speed of light (SPL) constraints.[4] Figure 33 depicts the CDF of latency inflation, which is defined as the

[4]We use $\frac{2}{3} * C$ within our calculations Singla et al. (2014)

ratio of measured latency and SPL latency calculated using line-of-sight distances for each CP.

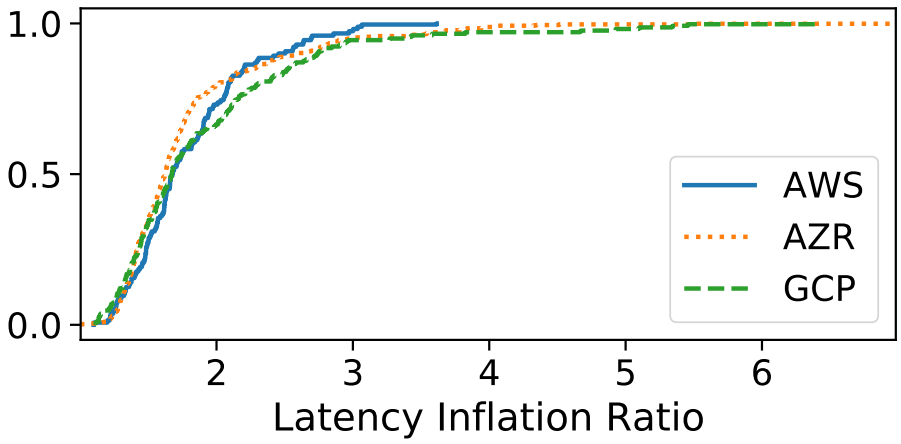


Figure 33. Distribution of latency inflation between network latency and RTT approximation using speed of light constraints for all regions of each CP.

We observe median latency inflation of about 1.68, 1.63, and 1.67 for intra-CP paths of AWS, Azure, and GCP, respectively. Compared to a median latency inflation ratio of 3.2 for public Internet paths Singla et al. (2014), these low latency inflation ratios attest to the optimal fiber paths and routes that are employed by CPs. Furthermore, Azure and GCP paths have long-tail in their latency inflation distributions while all intra-CP paths for AWS have a ratio of less than 3.6, making it the most optimal backbone among all CPs.

Main findings: *CPs employ an optimal fiber backbone with near line-of-sight latencies to create a global network. This result opens up a tantalizing opportunity to construct multi-cloud overlays in a performance-aware manner.*

6.3.2.3 Latency Characteristics of CP Backbones. Next, we turn our attention to the latency characteristics of the CP backbones toward the goal of creating CP-specific latency profiles. Figure 34 shows the distribution of RTT and standard deviation across different measurements for all paths between VM

pairs. We observe a wide range of RTT values between VM instances, which can be explained by the geographic distance between CP regions. Furthermore, latency between each pair is relatively stable across different measurements with a 90^{th}-percentile coefficient of variation of less than 0.05.

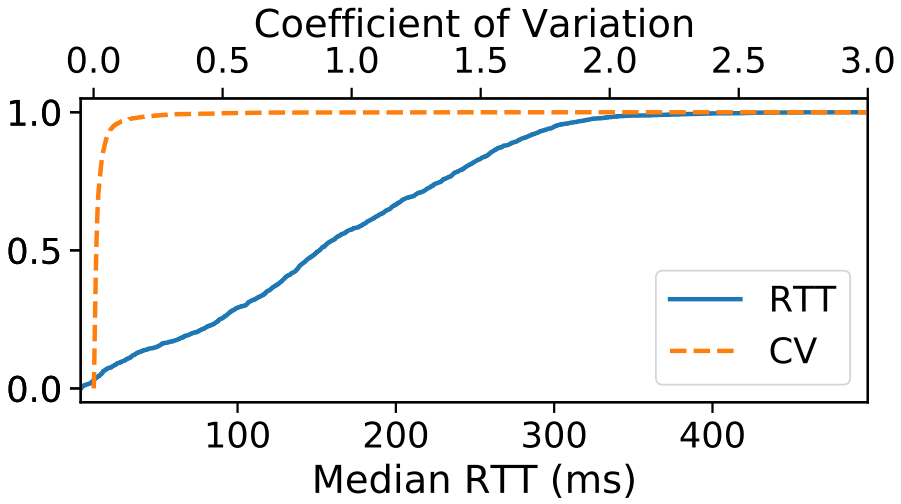


Figure 34. Distribution of median RTT and coefficient of variation for latency measurements between all VM pairs.

In addition to stability characteristics, we also compare the forward and reverse path latencies by measuring the difference between the median of latencies in each direction. We find that paths exhibit symmetric latencies with a 95^{th}-percentile latency difference of 0.22ms among all paths as shown in Figure 35.

Main findings: *Cloud paths exhibit a stable and symmetric latency profile over our measurement period, making them ideal for reliable multi-cloud overlays.*

6.3.3 Are Multi-Cloud Paths Better Than Single Cloud Paths?.

6.3.3.1 Overall Latency Improvements. The distribution of latency reduction percentage for all, intra-CP, and inter-CP paths is shown in Figure 36. From this figure, we observe that about 55%, 76%, and 69% of all, intra-CP, and inter-CP paths experience an improvement in their latency using an indirect

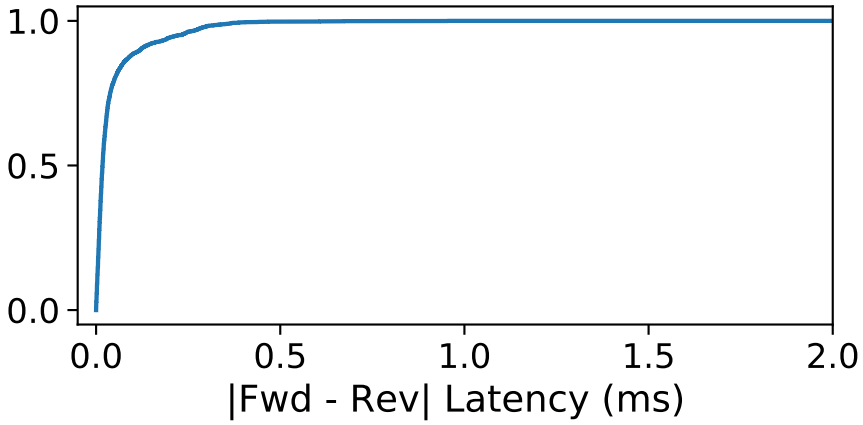


Figure 35. Distribution for difference in latency between forward and reverse paths for unique paths.

optimal path. These optimal paths can be constructed by relaying traffic through one or multiple intermediary CP regions. We provide more details on the intra- and inter-CP optimal overlay paths below.

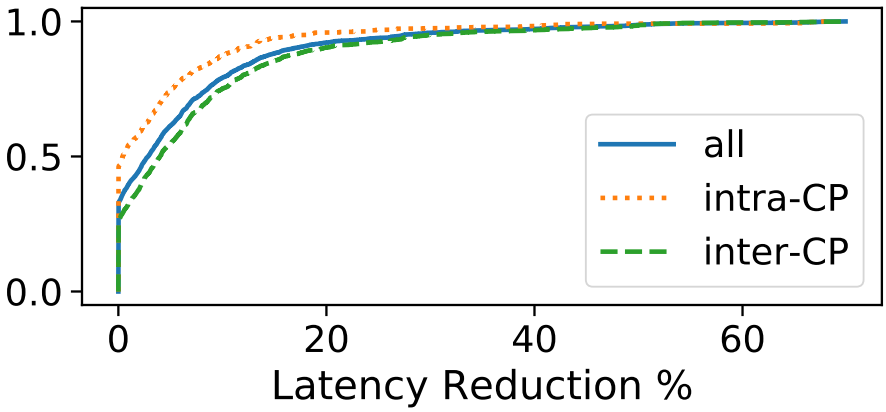


Figure 36. Distribution for RTT reduction ratio through all, intra-CP, and inter-CP optimal paths.

To complement Figure 36, Figure 37-(left) shows the distribution of the number of relay hops along optimal paths. From this figure, we find that the majority (64%) of optimal paths can be constructed using *only* one relay hop while some paths can go through as many as 5 relay hops. Almost all of the optimal

paths with latency reductions greater than 30% have less than 4 relay hops as shown in Figure 37-(right). In addition, we observe that the median of latency reduction percentage increases with the number of relay hops. We note that (a) forwarding traffic through additional relay hops might have negative effects (e.g., increase in latencies) and (b) optimal paths with many relay hops might have an alternative path with fewer hops and comparable performance.

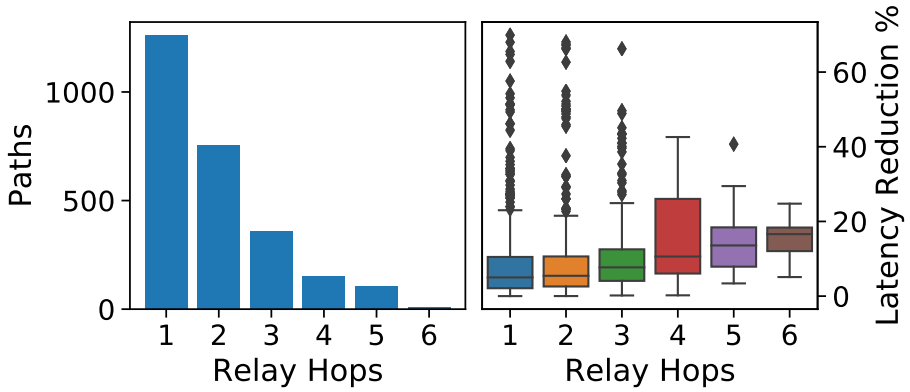


Figure 37. Distribution for the number of relay hops along optimal paths (left) and the distribution of latency reduction percentage for optimal paths grouped based on the number of relay hops (right).

Lastly, we measure the prevalence of each CP along optimal paths and find that AWS, Azure, and GCP nodes are selected as relays for 55%, 48%, and 28% of optimal paths.

6.3.3.2 Intra-CP Latency Improvements. We present statistics on the possibility of optimal overlay paths that are sourced and destined towards the same CP network (i.e. intra-CP overlays). Figure 38 depicts the distribution of latency reduction ratio for intra-CP paths of each CP. The distributions are grouped based on the CP network. Furthermore, each boxplot's color represents the ownership of relay nodes with *A, Z,* and *G* corresponding to AWS, Azure, and GCP relays respectively. From this figure, we observe that intra-CP paths

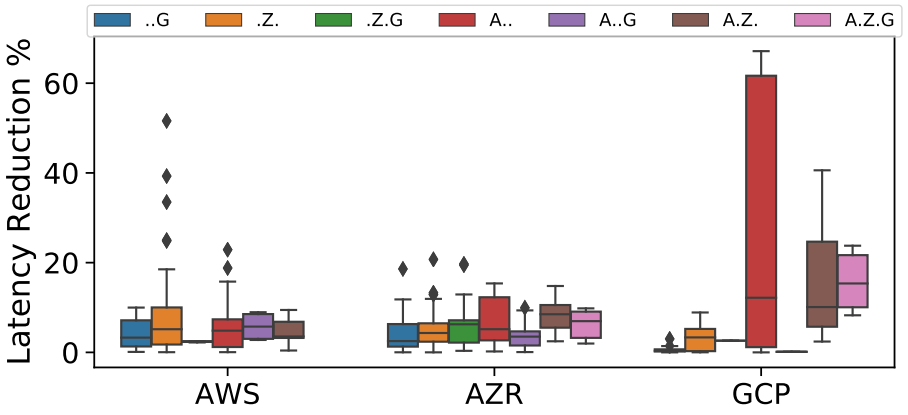


Figure 38. Distribution of latency reduction percentage for intra-CP paths of each CP, divided based on the ownership of the relay node.

can benefit from relay nodes within their own network in addition to nodes from other CPs. Furthermore, we observe that intra-CP paths within GCP's network observe the greatest reduction in latency among all CPs with AWS relays being the most effective in lowering the end-to-end latency. Upon closer examination, we observe that the majority of these paths correspond to GCP regions within Europe communicating with GCP regions in either India or Hong Kong.

Main findings: *Our measurements demonstrate that surprisingly, intra-CP paths can observe end-to-end latency reductions via optimal paths that are constructed with relay hops that belong to a different CP.*

6.3.3.3 Inter-CP Latency Improvements. We next focus on the possibility of overlay paths that are sourced from one CP but destined towards a different CP (i.e. inter-CP overlays). Figure 39 presents the latency reduction percentage for inter-CP paths. For brevity, only one direction of each CP pair is presented as the reverse direction is identical. Similar to Figure 38 the color and label encoding of each boxplot represent the ownership of relay nodes. From this figure, we make a number of observations. First, optimal paths constructed

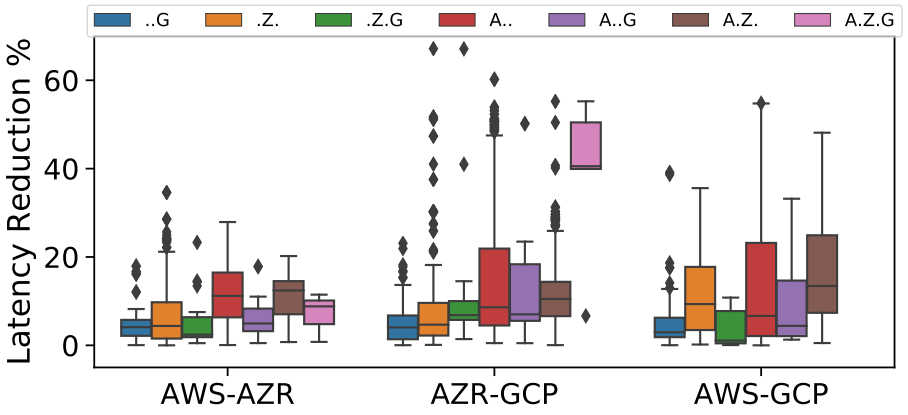


Figure 39. Distribution of latency reduction ratio for inter-CP paths of each CP, divided based on the ownership of the relay nodes.

using GCP nodes *as relays* exhibit the least amount of latency reduction. Second, *AWS-AZR* paths have lower values of latency reduction with equal amounts of reduction across each relay type. This is indicative of a tight coupling between these networks. Lastly, optimal paths with AWS relays tend to have higher latency reductions which are in line with our observations in §6.3.2.1 regarding AWS' backbone.

Main findings: *Similar to intra-CP paths, inter-CP paths can benefit from relay nodes to construct new, optimal paths with lower latencies. Moreover, inter-CP paths tend to experience greater reductions in their latency.*

6.3.4 Are there Challenges in Creating Multi-Cloud Overlays?.

6.3.4.1 Traffic Costs of CP Backbones. We turn our focus to the cost of sending traffic via CP backbones. Commonly, CPs charge their customers for traffic that is transmitted from their VM instances. That is, customers are charged *only* for egress traffic; all ingress traffic is free. Moreover, traffic is billed on a volume-by-volume basis (e.g., per GB of egress traffic) but each CP has a different set of rules and rates that govern their pricing policy. For example, we find

that AWS and GCP have lower rates for traffic that remains within their network (i.e. is sourced and destined between different regions of their network) while Azure is agnostic to the destination of the traffic. Furthermore, GCP has different rates for traffic destined to the Internet based on the geographic region of the destination address. We compile all these pricing policies based on the information that each CP provides on their webpage Amazon (2019c); Google (2019); Microsoft (2019) into a series of rules that allow us to infer the cost of transmitting traffic from each CP instance to other destinations.

Traffic costs for AWS. For AWS (see Figure 40), we observe that intra-CP traffic is always cheaper than inter-CP traffic with the exception of traffic that is sourced from Australia and Korea. Furthermore, traffic sourced from the US, Canada, and European regions have the lowest rate while traffic sourced from Brazil has the highest charge rate per volume of traffic. Lastly, traffic is priced in multiple tiers defined based on the volume of exchanged traffic and we see that exchanging extra traffic leads to lower charging rates.

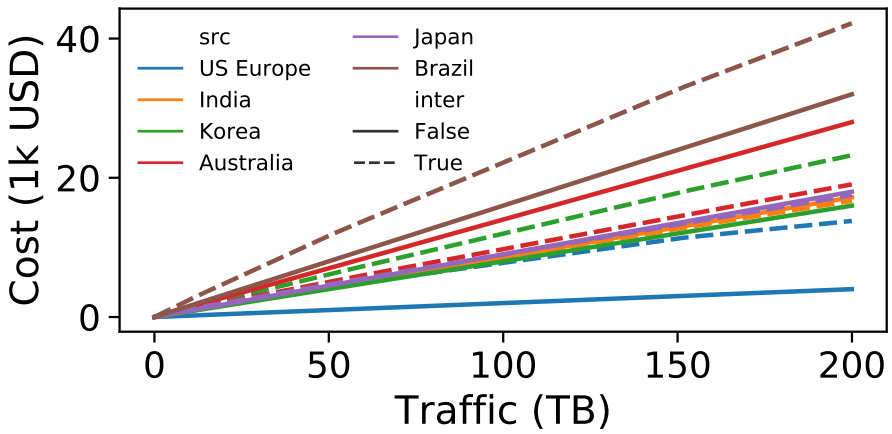


Figure 40. Cost of transmitting traffic sourced from different groupings of AWS regions. Dashed (solid) lines present inter-CP (intra-CP) traffic cost.

Traffic costs for Azure. Azure's pricing policy is much more simple (see Figure 41). Global regions are split into multiple large size areas namely (i) North America and Europe excluding Germany, (ii) Asia and Pacific, (iii) South America, and (iv) Germany. Each of these areas has a different rate, with North America and Europe being the cheapest while traffic sourced from South America can cost up to 3x more than North America. Lastly, as mentioned earlier, Azure is agnostic to the destination of traffic and does not differentiate between intra-CP and traffic destined to the Internet.

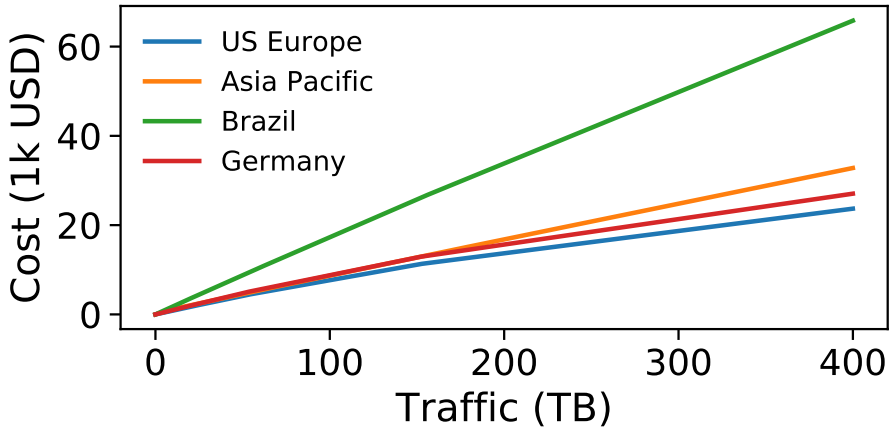


Figure 41. Cost of transmitting traffic sourced from different groupings of Azure regions.

Traffic costs for GCP. GCP's pricing policy is the most complicated among the top 3 CPs (see Figure 42). At a high level, GCP's pricing policy can be determined based on (i) source region, (ii) destination geographic location, and (iii) whether the destination is within GCP's network or the Internet (intra-CP vs inter-CP). Intra-CP traffic generally has a lower rate compared to inter-CP traffic. Furthermore, traffic destined to China (excluding Hong Kong) and Australia have higher rates compared to other global destinations.

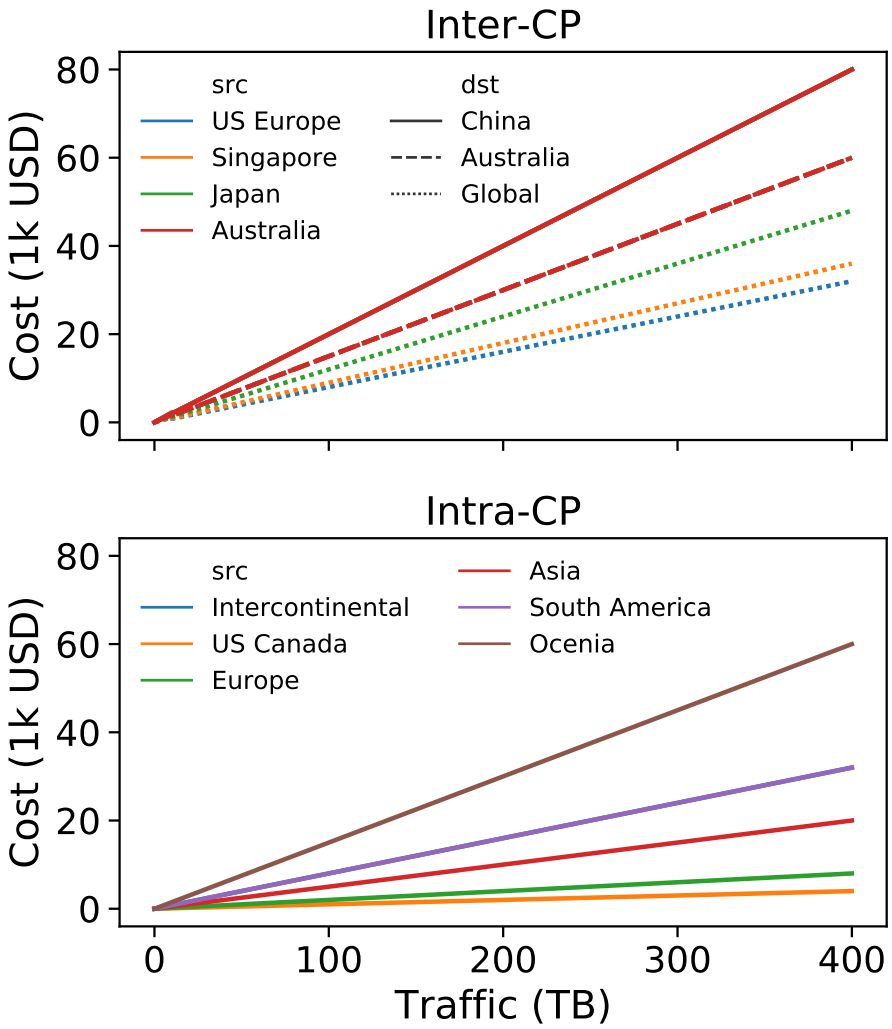


Figure 42. Cost of transmitting traffic sourced from different groupings of GCP regions. Solid, dashed, and dotted lines represent cost of traffic destined to China (excluding Hong Kong), Australia, and all other global regions accordingly.

6.3.5 Cost Penalty for Multi-Cloud Overlays. Next, we seek an answer to the question of the cost incurred by using relay nodes from other CPs. Figure 43 depicts the distribution of cost penalty (i.e. the difference between the optimal overlay cost and default path cost) within various latency reduction percentage bins for transmitting 1TB of traffic.

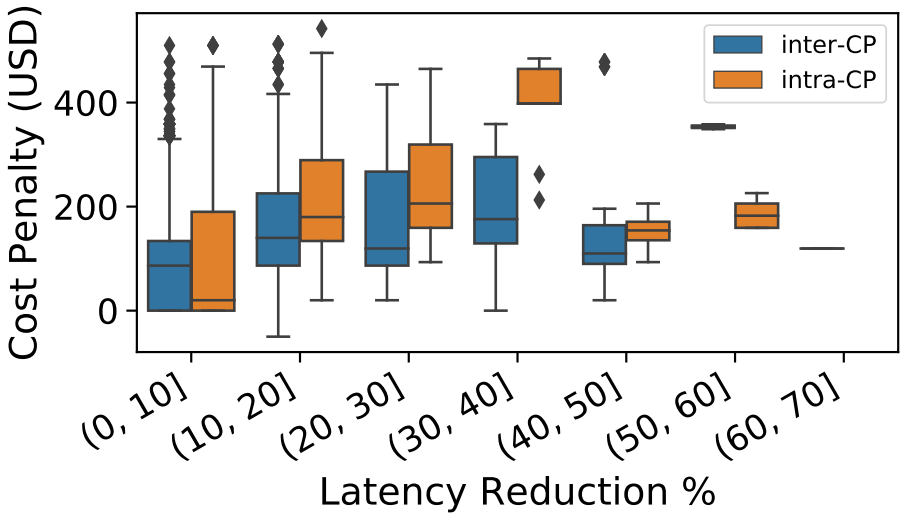


Figure 43. Distribution of cost penalty within different latency reduction ratio bins for intra-CP and inter-CP paths.

From Figure 43, we make a number of key observations. First, we find that optimal paths between intra-CP endpoints incur higher cost penalties compared to inter-CP paths. This is expected as intra-CP paths tend to have lower charging rates and optimal overlays usually pass through a 3rd party CP's backbone. Counter-intuitively, we next observe that the median cost penalty for paths with the most amount of latency reduction is less or equal to less optimal overlay paths. Lastly, we find that 2 of our optimal overlay paths have a negative cost penalty. That is, the optimal path costs are lesser than transmitting traffic directly between the endpoints. Upon closer inspection, we find that all of these paths are destined to the AWS Australia region and are sourced from GCP regions in Oregon US and Montreal, Canada, respectively. All of these paths benefit from AWS' lower transit cost towards Australia by handing off their traffic towards a nearby AWS region. Motivated by this observation, for each set of endpoint pairs we find the path with the minimum cost. We find that the cost of traffic sourced from all GCP regions (except for GCP Australia) and destined to AWS Australia can be reduced by 28%

by relying on AWS' network as a relay hop. These cost-optimal paths on average experience a 72% inflation in their latency.

Main findings: *The added cost of overlay networks is not highly prohibitive. In addition to the inherent benefits of multi-cloud settings, our results demonstrate that enterprises and cloud users can construct high-performance overlay networks atop multi-cloud underlays in a cost-aware manner.*

6.3.6 Further Optimization Through IXPs. Motivated by the observations within Kotronis et al. (2016), we investigate the possibility of creating optimal inter-CP paths via IXP relays. Using this approach an enterprise (or possibly a third-party relay service provider) would peer CPs at IXPs that have multiple CPs present and would relay traffic between their networks. We should note that the results presented in this section offer upper bounds on the amount of latency reduction and realization of these values in practice are dependent on several factors including (i) enterprise or a third-party entity should be present at IXP relay points and has to peer with the corresponding CPs, (ii) relay nodes should implement an address translation scheme since CPs would only route traffic to destination addresses within a peers address space, (iii) CPs could have restrictions on which portion of their network is reachable from each peering point and therefor a customers cloud traffic might not be routable to certain IXP relays.

Towards this goal, we gather a list of ∼20k IXP tenant interface addresses using CAIDA's aggregate IXP dataset CAIDA (2018) corresponding to 741 IXPs in total. We limit our focus to 143 IXPs which host more than one of our target CPs (i.e. an enterprise or third-party relay provider has the opportunity to peer with more than one CP). Given that IXP tenants can peer remotely, we only limit our focus to the interface addresses of CPs within an IXP and perform path and

latency probes using the same methodology described in § 6.3.1. We approximate the latency of each CP region towards each unique IXP by relying on the median of measured latencies.

We augment our connectivity graph by creating nodes for each IXP and place an edge between IXPs and the regions of each CP that is a tenant of that IXP. Furthermore, we annotate edges with their corresponding measured minimum latency. Using this augmented graph we measure the optimal overlay paths between CP region pairs. Out of the 4.56k path, about 3.21k (compared to 3.16k CP based overlays) can benefit from overlay paths that have IXP relay nodes. About 0.19k of the optimized overlay paths exclusively rely on IXP relays, i.e. CPs do not appear as relay nodes along the path. Figure 44 depicts the distribution of latency reduction percentage for optimal paths using CP relays, IXP relays, and a combination of CP and IXP relays. From this figure, we observe that IXP relays offer minimal improvement to multi-cloud overlay paths indicating that CP paths are extremely optimized and that CPs tend to leverage peering opportunities with other CPs when available. We should note that the results in this section explore a hypothetical relay service provider that only operates within IXPs that have more than one CP. Further improvements in multi-cloud connectivity via dark fiber paths between IXPs/colos hosting a single CP are part of future work that we would like to investigate.

6.4 Evaluation of *Tondbaz*

6.4.1 Case Studies of Optimal Paths.

Given the large number of possible paths between all CP regions, we select a handful of large scale areas that are most likely to be utilized by enterprises which have WAN deployments. For

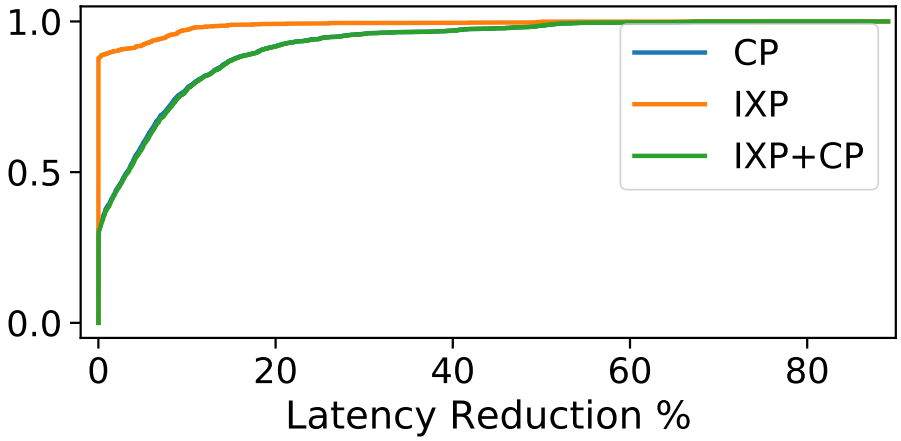


Figure 44. Distribution for RTT reduction percentage through CP, IXP, and CP+IXP relay paths.

each set of regions, we present the optimal path and discuss the cost penalty for traversing through this path.

US East - US West: All of our target CPs have representative regions near northern Virginia on the east coast of the US. Contrary to the east coast, CP regions on the west coast are not concentrated in a single area. AZR is the only CP with a region within the state of Washington, GCP and AWS have deployments within Oregon, and AWS and AZR have regions in northern California while GCP has a region in southern California. The shortest path between US coasts is possible through an overlay between AZR on the east coast and AWS in northern California with a median RTT of 59.1 ms and a traffic cost of about $86 for transmitting 1 TB of data. By accepting a 1 ms increase in RTT the cost of transmitting 1 TB of traffic can be reduced to $10 by utilizing GCP regions on both US coasts.

US East - Europe: For brevity, we group all european regions together. The optimal path between the east coast of US and Europe is between AWS in northern Virginia and AZR in Ireland with a median RTT of 66.4 ms and a traffic cost of

about $90 for transmitting 1 TB of data. The cost of traffic can be reduced to $20 for 1 TB of data by remaining within AWS' network and transmitting traffic between AWS in norther Virginia and AWS in Ireland with an RTT of 74.7 ms.

US East - South America: All CP regions within south America are located in São Paulo Brazil. The optimal path between these areas is established through AZR in northern Virginia and AWS in Brazil. The median of RTT for this path is 116.7 ms and transmitting 1 TB of data would cost about $87. Interestingly, this optimal path also has the lowest cost for transmitting data between these areas.

US East - South Africa: AZR is the only CP that is present in South Africa, the optimal path between each CP's region in northern Virginia and AZR's region in South Africa all have the same amount RTT of about 231 ms with the traffic cost for sourcing traffic from AZR, AWS, and GCP in northern Virginia is $86, $90, and $110 accordingly.

US West - South America: As stated earlier the CP regions on the west coast of US are not concentrated in a single area. The most optimal path from all CP regions on the west coast is between GCP in southern California and GCP in Brazil with an RTT of 167.5 ms with a traffic cost of $80 for exchanging 1 TB of data. The optimal path from northern California is made possible through AZR's region in northern California and AWS in Brazil with an RTT of 169.5 ms and a traffic cost of $86.5 for 1 TB of data. The cheapest path for exchanging 1 TB of traffic is made possible through AWS in northern California and AWS in Brazil for $20 with an RTT of 192.2 ms.

US West - Asia East: For east Asia, we consider regions within Japan, South Korea, Hong Kong, and Singapore. The optimal path between the west coast of US and east Asia is possible through GCP's region on US west coast and GCP in

Tokyo Japan with an RTT of 88.5 ms and a traffic cost of $80 for transmitting 1 TB of traffic. Optimal paths destined to AZR in Japan tend to go through GCP relays. The cheapest path for transmitting 1 TB of data from US west coast to east Asia is possible through AWS in Oregon and AWS in Japan for $20 and a median RTT of 98.6 ms.

US West - Australia: AWS and GCP both have regions within Sydney Australia while AZR has regions in Sydney, Canberra, and Melbourne Australia. The optimal path from US west coast towards Australia is sourced from GCP in southern California and GCP in Australia with a median RTT of 137 ms and a traffic cost of $150 for 1 TB of data. The next optimal path is possible through AWS in Oregon and AWS in Australia with a median RTT of 138.9 ms and a traffic cost of $20 for 1 TB of data. Optimal paths for other combinations of regions typically benefit from going through GCP and AWS relays with the latter option having lower traffic cost.

India - Europe: All 3 CPs have regions within Mumbai India. Furthermore, AZR has 2 more regions within India namely in Pune and Chennai. The optimal path from India towards Europe is sourced from AWS in Mumbai and destined to AWS in France with a median RTT of 103 ms and a traffic cost of $86 for 1 TB of data. The cheapest path is sourced from GCP India and destined to GCP in Belgium with a traffic cost of $80 for 1 TB of data and a median RTT of 110 ms.

6.4.2 Deployment of Overlays.

In this section, we demonstrate how *Tondbaz* creates multi-cloud overlays and empirically measure the latency reduction through the overlay and contrast them with *Tondbaz's* estimated latency reductions based on its internal model.

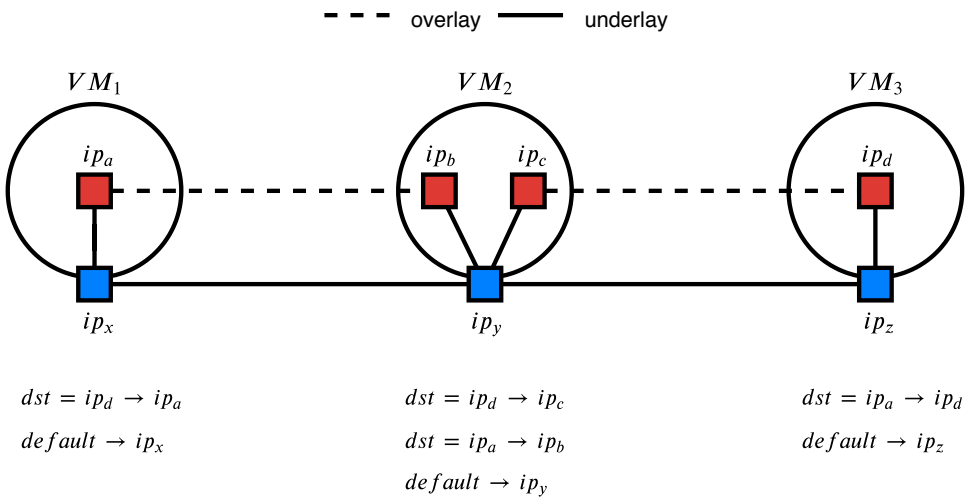


Figure 45. Overlay network composed of 2 nodes (VM_1 and VM_3) and 1 relay node (VM_2). Forwarding rules are depicted below each node.

Network overlays can be created either at the application layer or happen transparently at the network layer. In the former case, each application is responsible for incorporating the forwarding logic into the program while in the latter case applications need not be aware of the forwarding logic within the overlay and simply need to utilize IP addresses within the overlay domain. Given the wide range of applications that could be deployed within a cloud environment, we chose to create multi-cloud overlays at the network level.

The construction of overlays consists of several high-level steps, namely (i) identifying a private overlay subnet which does not overlap with the private address space of participating nodes, (ii) assigning unique IP addresses to each overlay node including relay nodes, (iii) creating virtual tunneling interfaces and assigning their next-hop address based on the inferred optimal overlay path, and (iv) creating forwarding rules for routing traffic through the correct tunneling interface.

To illustrate these steps consider the example overlay network in Figure 45 composed of two nodes (VM_1 and VM_3) and one relay node (VM_2). Each

node has a default interface (highlighted in blue) that is connected to the public Internet. Furthermore, each node can have one or two virtual tunneling interfaces depending on whether they are a regular or relay node in the overlay respectively. Below each node, the forwarding rules to support the overlay network are given. Based on the given forwarding rules, a data packet sourced from an application on VM_1 destined to ip_d on VM_3 would be forwarded to interface ip_a where the packet would be encapsulated inside an additional IP header and forwarded to ip_y on VM_2. Upon the receipt of this packet, VM_2 would decapsulate the outer IP header and since the packet is destined to ip_d, it would be forwarded to ip_c where it would be encapsulated once again inside an IP header destined to ip_z. Once VM_3 receives the packet, it would decapsulate the outer IP header and would forward the data packet to the corresponding application on VM_3.

Initially, we implemented the overlay construction mechanism using the IPIP module of Linux which simply encapsulates packets within an IP header without applying any encryption to the payload. Although we were able to establish overlay tunnels within GCP and AWS' network, for an unknown reason Azure's network would drop our tunneled packets. For this reason, we migrated our tunneling mechanism to WireGuard WireGuard (2019) which encrypts the payload and encapsulates the encrypted content within an IP+UDP header. This encapsulation mechanism has a minimum of 28 Bytes of overhead corresponding to 8 Bytes for the UDP header + a minimum of 20 Bytes for the IP header which translates to less than 2% overhead for a 1500 MTU.

6.4.2.1 Empirical vs Estimated Overlay Latencies. As stated earlier given the large number of possibilities for creating overlay networks we limit our focus to a handful of cases where *Tondbaz* estimated a reduction in end-to-end

Table 9. List of selected overlay endpoints (first two columns) along the number of relay nodes for each overlay presented in the third column. The default RTT, estimated overlay RTT, and empirical RTT are presented in the last three columns respectively.

source	**destination**	**relays**	**default RTT (ms)**	**overlay RTT (ms)**	**empiric RTT (ms)**	**RTT saving (ms)**
AWS Hong Kong	GCP Hong Kong	1	15.79	2.14	2.25	13.54
AZR Wyoming	GCP Oregon	1	49.91	33.74	33.86	16.05
GCP India	GCP Germany	2	351.5	148.8	149.02	202.48
GCP Singapore	AZR UAE	3	250.08	83.56	84.42	165.66

latency through an overlay network. Table 9 lists the set of selected end-points, number of relay nodes, the RTT of the default path, and *Tondbaz's* estimated RTT through the optimal overlay. While limited in number, the selected overlay paths represent a different combination of CP networks, geographic regions, number of relays, and latency reductions. Additionally, we list the set of relay nodes for each selected end-points within Table 10.

Table 10. List of selected overlay endpoints (first two columns) along with the optimal relay nodes (third column).

source	**destination**	**relays**
AWS Hong Kong	GCP Hong Kong	AZR Hong Kong
AZR Wyoming	GCP Oregon	AZR Washington
GCP India	GCP Germany	AWS India - AWS Germany
GCP Singapore	AZR UAE	AZR Singapore - AZR S.India - AZR W.India

For each overlay network, we conduct latency probes over the default and overlay paths for the full duration of a day using 5 minute rounds. Within each round we send 5 latency probes towards each destination address, resulting in a total of about 1.4k measurement samples per endpoint. Additionally, we also probe each VM's default interface address to obtain a baseline of latency that is needed to traverse the network stack on each VM node. Similar to our observations in § 6.3.2, the measured latencies exhibit tight distributions over both default and overlay paths with a coefficient of variation of less than 0.06. The last column in Table 9 presents the median of empirical latency over the overlay paths. For all overlay paths, we observe that *Tondbaz's* estimate deviates less 1ms from empirical measures, with paths having a greater number of relays exhibiting larger amounts of deviation. The observed deviation values are inline with our estimates of network stack traversal overhead for each VM (median of 0.1ms).

Summary: in this section, we demonstrated Tondbaz's overlay construction strategy and showcased its applicability of it through the construction of 4 optimal overlays. Although limited in number, these overlays exhibit the accuracy of Tondbaz's internal model in assessing overlay end-to-end latency.

6.5 Summary

Market push indicates that the future of enterprises is multi-cloud. Unfortunately, there is a technology pull: what is critically lacking is a framework for seamlessly gluing the public cloud resources together in a cost- and performance-aware manner. A key reason behind this technology pull is the lack of understanding of the path, delay, and traffic-cost characteristics of CPs' private backbones. In this chapter, we presented *Tondbaz* as a cloud-centric measurement platform and decision support framework for multi-cloud

environments. We demonstrate the applicability of our framework by deploying on global cloud regions of AWS, Azure, and GCP. Our cloud-centric measurement study sheds light on the characteristics of CPs' (private) backbones and reveals several new/interesting insights including optimal cloud backbones, lack of delay and path asymmetries in cloud paths, possible latency improvements in inter- and intra-cloud paths, and traffic-cost characteristics. We present recommendations regarding optimal inter-CP paths for select geographic region pairs. Lastly, we construct a handful of overlay networks and empirically measure the latency through the overlay network and contrast our measures with *Tondbaz's* internal model.

CHAPTER VII

CONCLUSIONS & FUTURE WORK

7.1 Conclusions

Cloud providers have been transformative to how enterprises conduct their business. By virtualizing vast resources of compute and storage through centralized data-centers, cloud providers have been an attractive alternative to maintaining in-house infrastructure and well adopted by private and public sectors. While cloud resources have been the center of many research studies, little attention has been dealt towards the connectivity of cloud providers and their effect on the topological structure of the Internet. In this dissertation we presented a holistic analysis of cloud providers and their role in today's Internet and made the following conclusions:

- Cloud providers in conjunction with CDN's are the major content providers in the Internet and collectively are responsible for a significant portion of an edge networks traffic;
- Similar to CDN networks, cloud providers have been making efforts to reduce their network distance by expanding the set of centralized compute regions in addition to offering new peering services (VPIs) to edge networks;
- In terms of connectivity of an enterprise towards cloud providers many factors including the type of connectivity (CPP, TPP, and BEP), cloud providers routing strategy, geo-proximity of cloud resources, and cross-traffic and congestion of TPP networks should be taken into consideration;

– The optimal backbone of cloud providers in combination with the tight interconnectivity of cloud provider networks with each other can be leveraged towards the creation of optimal overlays that have a global span;

Specifically, we have made the following contributions in each chapter.

In Chapter III we utilized traffic traces from an edge network (UOnet) to study the traffic footprint of major content providers and more specifically outline the degree to which their content is served from nearby locations. We demonstrated that the majority of traffic is associated with CDN and cloud providers networks. Furthermore, we devised a technique to identify cache servers residing within other networks which further enlarging the share of CDN networks towards traffic. Lastly, we quantify the effects of content locality on user-perceived performance and observe that many other factors such as last-mile connectivity are the main bottlenecks of performance for end-users.

In Chapter IV we present a measurement study of the interconnectivity fabric of Amazon as the largest cloud provider. We pay special attention to VPIs as an emergent and increasingly popular interconnection option for entities such as enterprises that desire highly elastic and flexible connections to cloud providers which bypass the public Internet. We present a methodology for capturing VPIs and offer lower bound estimates on the number of VPI peerings that Amazon utilizes. Next, we present a methodology for geolocating both ends of our inferred peerings. Lastly, we characterize customer networks that peer over various peering options (private, public, VPI) and offer insight into the visibility and routing implications of each peering type from the cloud providers' perspective.

In Chapter V we perform a third-party measurement study to understand the tradeoffs between three multi-cloud connectivity options (CPP, TPP, and

BEP). Based on our cloud-centric measurements, we find that CPP routes are better than TPP routes in terms of latency as well as throughput. We attribute the observed performance benefits to CPs' rich connectivity with other CPs and CPs' stable and well-designed private backbones. Additionally, we characterize the routing strategies of CPs (hot- cold- potato routing) and highlight their implications on end-to-end network performance metrics. Lastly, we identify that subpar performance characteristics of TPP routes are caused by several factors including border routers, queuing delays, and higher loss-rates on these paths.

In Chapter VI we propose and design *Tondbaz* as a measurement platform and decision support framework for multi-cloud settings. We demonstrate its applicability by conducting path and latency measurements between the global regions of AWS, Azure, and GCP networks. Our measurements highlight the tight interconnectivity of cloud providers networks on a global scale with backbones offering reliable connectivity to their customers. We utilize *Tondbaz* to measure optimal cloud overlays between various endpoints and by establishing traffic cost models for each cloud provider and inputting them to the decision support framework of *Tondbaz* we offer insight into the tradeoffs of cost vs performance. Next, we offer recommendations regarding the best connectivity paths between various geographic regions. Lastly, we deploy a handful of overlay networks and through empirical measurements, demonstrate the accuracy of *Tondbaz's* network performance estimates based on its internal model.

7.2 Future Work

In the following, we present several possible directions for future work that are in line with the presented dissertation.

– Exploring the possibility of further improving the connectivity of multi-cloud paths via the utilization of dark fiber links either by cloud providers or third-party connectivity providers is an open research problem. Investigating these possibilities can be beneficiary in obtaining improved multi-cloud connectivity in addition to improving the connectivity of poorly connected cloud regions;

– Complementary to our work in Chapter VI, one could measure and profile the connectivity and performance of edge networks towards cloud providers. Profiling the last mile of connectivity between edge users and cloud providers is equally important to study of cloud providers' backbone performance. This study in conjunction with the optimal cloud overlays generated by *Tondbaz* would enable us to provide estimates on the performance characteristics of connectivity between edge-users which is facilitated via optimal cloud overlays. In light of the rapid expansion of cloud providers backbones and their increasing role in the transit of Internet traffic conducting this study is of high importance and can provide insight into possible directions of end-user connectivity;

– In the current state of the Internet end-users are accustomed to utilizing many free services such as email, video streaming, social media networks, etc. The majority of these free services are funded via targeted advertisement platforms that rely on constructing accurate profiles of users based on their personal interests. These Internet services are based on an economic model of exchanging a user's personal data and time in return for utilizing free services. The past years have seen an increased interest in the development and adoption of decentralized alternatives Calendar (2019); Docs (2019); Fediverse (2019); Forms.id (2019); IPFS (2019); Mastodon (2019); PeerTube

(2019) for many Internet services. These decentralized applications rely on strong cryptography to ensure that only users with proper access/keys have access to the data. Furthermore, given their decentralized nature, the governance of data is not in the hands of a single entity. Decentralized or P2P services can have varying performance depending on the state of the network. The constant push of cloud providers for increasing their locality to end-users in conjunction with the vast amount of storage and compute resources within cloud regions makes them an ideal candidate for having a hybrid deployment of these decentralized services, where part of the deployment is residing on cloud regions and the remainder is deployed on end-user. Studying the performance of decentralized services in a hybrid deployment and contrasting it with their centralized counterparts would facilitate the wide adoption of these services by end-users. Furthermore, estimating the per user operational cost of running these services within cloud environments could be helpful in the advocacy of democratized Internet services;

– The rise in multi-cloud deployments by enterprises has fueled the emergence/expansion of cloud providers as well as third-party connectivity providers. The stakeholders in a multi-cloud setting including cloud providers, third-party connectivity providers, and enterprises can have incongruent goals or objectives. For example, cloud providers are interested in maximizing their profit by following certain routing policies while an enterprise is interested in maximizing their performance for the lowest operational cost via the adoption of multi-cloud overlays. Furthermore, stakeholders could lack incentive for sharing information retaining to their internal operation with each other. For example, cloud providers host applications on a set of heterogeneous hardware

which in turn could introduce varying degrees of performance for enterprises. Measuring, modeling and mitigating the tussles between all stakeholders of a multi-cloud ecosystem is crucial for the advancement of multi-cloud deployments.

– The optimal overlays outlined in Chapter VI would only be beneficial to enterprises that maintain and manage their compute resources, i.e. they do not rely on the added/managed services that cloud providers offer. For example, an enterprise can maintain its stream processing pipeline using Apache Kafka within their cloud instances or rely on managed services like Amazon MSK Amazon (2019a) or Confluent for GCP users Google (2019). In the former case given that an enterprise is in complete control of the service they can benefit from the overlays that are constructed with *Tondbaz* while in the later case the network connectivity paths are maintained by the cloud provider. The seamless operation of these managed services in a multi-cloud setting would require the development of interoperability layers between managed services of cloud providers. Furthermore, the optimal operation of these managed services requires additional APIs that expose the network layer and provide finer control to cloud users.

– Evaluating the connectivity performance for various third-party connectivity providers (TPPs) and a push for the disclosure of such information via public measurement platforms would be beneficial for enterprises seeking optimal hybrid or multi-cloud deployments;

– Exploring the adoption of VPIs by the customers of other cloud providers, in addition to repeating the measurements outlined in Chapter IV on a

temporal basis would offer a more comprehensive picture of Internet topology in addition to capturing the micro-dynamics of Internet peering enabled by VPIs;

REFERENCES CITED

Adhikari, V. K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., & Zhang, Z.-l. (2012). Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery. In *INFOCOM.* IEEE.

Ager, B., Chatzis, N., Feldmann, A., Sarrar, N., Uhlig, S., & Willinger, W. (2012). Anatomy of a large european ixp. *SIGCOMM CCR.*

Ager, B., Mühlbauer, W., Smaragdakis, G., & Uhlig, S. (2011). Web content cartography. In *Internet Measurement Conference (IMC).* ACM.

Akamai. (2017). *Akamai Technologies Facts & Figures.* https://www.akamai.com/us/en/about/facts-figures.jsp.

Alexander, M., Luckie, M., Dhamdhere, A., Huffaker, B., KC, C., & Jonathan, S. M. (2018). Pushing the boundaries with bdrmapit: Mapping router ownership at internet scale. In *Internet Measurement Conference (IMC).*

Amazon. (2018a). *AWS Direct Connect.* `https://aws.amazon.com/directconnect/`.

Amazon. (2018b). *AWS Direct Connect Frequently Asked Questions.* `https://aws.amazon.com/directconnect/faqs/`.

Amazon. (2018c). *AWS Direct Connect Partners.* `https://aws.amazon.com/directconnect/partners/`.

Amazon. (2018d). *AWS Direct Connect | Product Details.* `https://aws.amazon.com/directconnect/details/`.

Amazon. (2018e). *Describe virtual interfaces.* `https://docs.aws.amazon.com/cli/latest/reference/directconnect/describe-virtual-interfaces.html`.

Amazon. (2018f). *Regions and Availability Zones - Amazon Elastic Compute Cloud.* `https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-regions-availability-zones`.

Amazon. (2019a). *Amazon Managed Streaming for Apache Kafka.* `https://aws.amazon.com/msk/`.

Amazon. (2019b). *AWS Transit Gateway.* `https://aws.amazon.com/transit-gateway/`.

Amazon. (2019c). *EC2 Instance Pricing.* `https://aws.amazon.com/ec2/pricing/on-demand/`.

Andersen, D., Balakrishnan, H., Kaashoek, F., & Morris, R. (2001). Resilient overlay networks. In *SOSP.* ACM.

Anwar, R., Niaz, H., Choffnes, D., Cunha, Í., Gill, P., & Katz-Bassett, E. (2015). Investigating interdomain routing policies in the wild. In *Internet Measurement Conference (IMC).*

APNIC. (2018). *Measuring IPv6.* `https://labs.apnic.net/measureipv6/`.

Augustin, B., Friedman, T., & Teixeira, R. (2007). Multipath tracing with paris traceroute. In *End-to-End Monitoring Techniques and Services.* IEEE.

Augustin, B., Krishnamurthy, B., & Willinger, W. (2009). IXPs: Mapped? In *Internet Measurement Conference (IMC).* ACM.

Baker, F. (1995). *Requirements for IP Version 4 Routers* (Tech. Rep.). Cisco Systems.

Bender, A., Sherwood, R., & Spring, N. (2008). Fixing ally's growing pains with velocity modeling. In *SIGCOMM.* ACM.

Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., ... Seskar, I. (2014). GENI: A federated testbed for innovative network experiments. *Computer Networks*.

Beverly, R. (2016). Yarrp'ing the internet: Randomized high-speed active topology discovery. In *Internet Measurement Conference (IMC).* ACM.

Beverly, R., Durairajan, R., Plonka, D., & Rohrer, J. P. (2018). In the IP of the beholder: Strategies for active IPv6 topology discovery. In *Internet Measurement Conference (IMC).* ACM.

Böttger, T., Cuadrado, F., Tyson, G., Castro, I., & Uhlig, S. (2016). Open connect everywhere: A glimpse at the internet ecosystem through the lens of the netflix cdn. *arXiv preprint arXiv:1606.05519*.

Bozkurt, I. N., Aqeel, W., Bhattacherjee, D., Chandrasekaran, B., Godfrey, P. B., Laughlin, G., ... Singla, A. (2018). Dissecting latency in the internet's fiber infrastructure. *arXiv preprint arXiv:1811.10737*.

Build Azure. (2019). *Microsoft Azure Region Map.* `https://map.buildazure.com/`.

Burrington, I. (2016). *Why Amazon's Data Centers Are Hidden in Spy Country.* https://www.theatlantic.com/technology/archive/2016/01/amazon-web-services-data-center/423147/.

CAIDA. (2018). *Archipelago (Ark) measurement infrastructure.* http://www.caida.org/projects/ark/.

CAIDA. (2018). *AS Relationships.* http://www.caida.org/data/as-relationships/.

CAIDA. (2018). *The CAIDA UCSD IXPs Dataset.* http://www.caida.org/data/ixps.xml.

Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., & Govindan, R. (2013). Mapping the Expansion of Google's Serving Infrastructure. In *Internet measurement conference (imc).*

Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., & Padhye, J. (2015). Analyzing the Performance of an Anycast CDN. In *Internet Measurement Conference (IMC).* ACM.

Calder, M., Gao, R., Schröder, M., Stewart, R., Padhye, J., Mahajan, R., ... Katz-Bassett, E. (2018). Odin: Microsoft's Scalable Fault-Tolerant {CDN} Measurement System. In *NSDI.* USENIX.

Calendar, S. (2019). *Secure Calendar - Free Encrypted Calendar.* https://securecalendar.online/.

Castro, I., Cardona, J. C., Gorinsky, S., & Francois, P. (2014). Remote Peering: More Peering without Internet Flattening. *CoNEXT*.

Chabarek, J., & Barford, P. (2013). What's in a name?: decoding router interface names. In *Hotplanet.* ACM.

Chandrasekaran, B., Smaragdakis, G., Berger, A. W., Luckie, M. J., & Ng, K.-C. (2015). A server-to-server view of the internet. In *Conext.* ACM.

Chatzis, N., Smaragdakis, G., Böttger, J., Krenc, T., & Feldmann, A. (2013). On the Benefits of Using a Large IXP as an Internet Vantage Point. In *Internet Measurement Conference (IMC).* ACM.

Chiu, Y.-C., Schlinker, B., Radhakrishnan, A. B., Katz-Bassett, E., & Govindan, R. (2015). Are we one hop away from a better internet? In *Internet Measurement Conference (IMC).* ACM.

Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., & Bowman, M. (2003). Planetlab: an overlay testbed for broad-coverage services. *SIGCOMM CCR*.

Comarela, G., Terzi, E., & Crovella, M. (2016). Detecting unusually-routed ASes: Methods and applications. In *Internet Measurement Conference (IMC).* ACM.

CoreSite. (2018). *The Coresite Open Cloud Exchange - One Connection. Countless Cloud Options.* `https://www.coresite.com/solutions/cloud-services/open-cloud-exchange`.

Costa, P., Migliavacca, M., Pietzuch, P., & Wolf, A. L. (2012). Naas: Network-as-a-service in the cloud. In *Workshop on hot topics in management of internet, cloud, and enterprise networks and services.*

Cunha, Í., Marchetta, P., Calder, M., Chiu, Y.-C., Machado, B. V., Pescapè, A., ... Katz-Bassett, E. (2016). Sibyl: a practical internet route oracle. *NSDI*.

DatacenterMap. (2018). *Amazon EC2.* `http://www.datacentermap.com/cloud/amazon-ec2.html`.

Demchenko, Y., Van Der Ham, J., Ngo, C., Matselyukh, T., Filiposka, S., de Laat, C., & Escalona, E. (2013). Open cloud exchange (OCX): Architecture and functional components. In *Cloud Computing Technology and Science.* IEEE.

Dhamdhere, A., Clark, D. D., Gamero-Garrido, A., Luckie, M., Mok, R. K., Akiwate, G., ... Claffy, K. (2018). Inferring persistent interdomain congestion. In *SIGCOMM.* ACM.

Dhamdhere, A., & Dovrolis, C. (2010). The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh. In *CoNEXT.* ACM.

Diamantidis, N., Karlis, D., & Giakoumakis, E. A. (2000). Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence*.

Docs, A. (2019). *Arcane Docs – Blockchain-based alternative for Google Docs.* `https://docs.arcaneoffice.com/signup/`.

Durairajan, R., Barford, C., & Barford, P. (2018). Lights Out: Climate Change Risk to Internet Infrastructure. In *Proceedings of the applied networking research workshop.* ACM.

Durairajan, R., Ghosh, S., Tang, X., Barford, P., & Eriksson, B. (2013). Internet atlas: a geographic database of the internet. In *Hotplanet.*

Durairajan, R., Sommers, J., & Barford, P. (2014). Layer 1-Informed Internet Topology Measurement. In *Internet Measurement Conference (IMC).* ACM.

Durairajan, R., Sommers, J., Willinger, W., & Barford, P. (2015). InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure. In *SIGCOMM.* ACM.

Durumeric, Z., Wustrow, E., & Halderman, J. A. (2013). Zmap: Fast internet-wide scanning and its security applications. In *Usenix security symposium.*

Eclipse. (2019). *Eclipse Paho - MQTT and MQTT-SN Software.* `http://www.eclipse.org/paho/`.

EdgeConneX. (2018). *Space, power and connectivity.* http://www.edgeconnex.com/company/about/.

Engebretson, J. (2014). *Verizon-netflix dispute: Is netflix using direct connections or not?* `https://www.telecompetitor.com/verizon-netflix-dispute-netflix-using-direct-connections/`.

The enterprise deployment game-plan: why multi-cloud is the future. (2018). `https://blog.ubuntu.com/2018/08/30/the-enterprise-deployment-game-plan-why-multi-cloud-is-the-future`.

Equinix. (2017). *Cloud Exchange.* `http://www.equinix.com/services/interconnection-connectivity/cloud-exchange/`.

Eriksson, B., Durairajan, R., & Barford, P. (2013). RiskRoute: A Framework for Mitigating Network Outage Threats. *CoNEXT*. doi: 10.1145/2535372.2535385

European Internet Exchange Association. (2018). `https://www.euro-ix.net/`.

Example Applications Services. (2018). `https://builtin.com/cloud-computing/examples-applications-services`.

Fan, X., Katz-Bassett, E., & Heidemann, J. (2015). Assessing Affinity Between Users and CDN Sites. In *Traffic monitoring and analysis.* Springer.

Fanou, R., Francois, P., & Aben, E. (2015). On the diversity of interdomain routing in Africa. In *Passive and active measurements (pam).*

Fediverse. (2019). *Fediverse.* `https://fediverse.party/`.

Five Reasons Why Multi-Cloud Infrastructure is the Future of Enterprise IT. (2018). `https://www.cloudindustryforum.org/content/five-reasons-why-multi-cloud-infrastructure-future-enterprise-it`.

Fontugne, R., Pelsser, C., Aben, E., & Bush, R. (2017). Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. In *Internet Measurement Conference (IMC).* ACM.

Fontugne, R., Shah, A., & Aben, E. (2018). The (thin) bridges of as connectivity: Measuring dependency using as hegemony. In *Passive and Active Measurement (PAM)*. Springer.

Forms.id. (2019). *Private, simple, forms. | Forms.id.* https://forms.id/.

The Future of IT Transformation Is Multi-Cloud. (2018). https://searchcio.techtarget.com/Rackspace/The-Future-of-IT-Transformation-Is-Multi-Cloud.

The Future of Multi-Cloud: Common APIs Across Public and Private Clouds. (2018). https://blog.rackspace.com/future-multi-cloud-common-apis-across-public-private-clouds.

The Future of the Datacenter is Multicloud. (2018). https://www.nutanix.com/2018/11/01/future-datacenter-multicloud/.

Gartner. (2016). https://www.gartner.com/doc/3396633/market-trends-cloud-adoption-trends.

Gasser, O., Scheitle, Q., Foremski, P., Lone, Q., Korczyński, M., Strowes, S. D., ... Carle, G. (2018). Clusters in the expanse: Understanding and unbiasing IPv6 hitlists. In *Internet Measurement Conference (IMC)*. ACM.

Gehlen, V., Finamore, A., Mellia, M., & Munafò, M. M. (2012). Uncovering the big players of the web. In *Lecture notes in computer science*. Springer.

Gharaibeh, M., Shah, A., Huffaker, B., Zhang, H., Ensafi, R., & Papadopoulos, C. (2017). A look at router geolocation in public and commercial databases. In *Internet Measurement Conference (IMC)*. ACM.

Gill, P., Arlitt, M., Li, Z., & Mahanti, A. (2008). The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse? In *Passive and Active Measurement (PAM)*. Springer.

Giotsas, V., Dhamdhere, A., & Claffy, K. C. (2016). Periscope: Unifying looking glass querying. In *Passive and Active Measurement (PAM)*. Springer.

Giotsas, V., Dietzel, C., Smaragdakis, G., Feldmann, A., Berger, A., & Aben, E. (2017). Detecting peering infrastructure outages in the wild. In *SIGCOMM*. ACM.

Giotsas, V., Luckie, M., Huffaker, B., & Claffy, K. (2015). Ipv6 as relationships, cliques, and congruence. In *Passive and Active Measurements (PAM)*.

Giotsas, V., Luckie, M., Huffaker, B., et al. (2014). Inferring Complex AS Relationships. In *Internet Measurement Conference (IMC)*. ACM.

Giotsas, V., Smaragdakis, G., Huffaker, B., Luckie, M., & claffy, k. (2015). Mapping Peering Interconnections to a Facility. In *CoNEXT*.

Giotsas, V., & Zhou, S. (2012). Valley-free violation in internet routing—analysis based on bgp community data. In *International conference on communications*.

Giotsas, V., & Zhou, S. (2013). Improving the discovery of IXP peering links through passive BGP measurements. In *INFOCOM*.

Giotsas, V., Zhou, S., Luckie, M., & Klaffy, K. (2013). Inferring Multilateral Peering. In *CoNEXT*. ACM.

Google. (2018a). *GCP Direct Peering.* `https://cloud.google.com/interconnect/docs/how-to/direct-peering`.

Google. (2018b). *Google supported service providers.* `https://cloud.google.com/interconnect/docs/concepts/service-providers`.

Google. (2018c). *Partner Interconnect | Google Cloud.* `https://cloud.google.com/interconnect/partners/`.

Google. (2019). *Apache Kafka for GCP Users.* `https://cloud.google.com/blog/products/gcp/apache-kafka-for-gcp-users-connectors-for-pubsub-dataflow-and-bigquery`.

Google. (2019). *Data center locations.* `https://www.google.com/about/datacenters/inside/locations/index.html`.

Google. (2019). *Google Compute Engine Pricing.* `https://cloud.google.com/compute/pricing#network`.

Govindan, R., & Tangmunarunkit, H. (2000). Heuristics for Internet map discovery. In *INFOCOM*.

Graham, R., Mcmillan, P., & Tentler, D. (2014). Mass Scanning the Internet: Tips, Tricks, Results. In *Def Con 22*.

Green, T., Lambert, A., Pelsser, C., & Rossi, D. (2018). Leveraging inter-domain stability for bgp dynamics analysis. In *Passive and Active Measurement (PAM)*. Springer.

Gregori, E., Improta, A., Lenzini, L., & Orsini, C. (2011). The impact of IXPs on the AS-level topology structure of the Internet. *Computer Communications*.

Gunes, M., & Sarac, K. (2009). Resolving IP aliases in building traceroute-based Internet maps. *Transactions on Networking (ToN)*.

Gunes, M. H., & Sarac, K. (2006). Analytical IP alias resolution. In *International conference on communications.*

Gupta, A., Calder, M., Feamster, N., Chetty, M., Calandro, E., & Katz-Bassett, E. (2014). Peering at the Internet's Frontier: A First Look at ISP Interconnectivity in Africa. *Passive and Active Measurements (PAM).*

Haq, O., Raja, M., & Dogar, F. R. (2017). Measuring and improving the reliability of wide-area cloud paths. In *WWW.* ACM.

He, Y., Siganos, G., Faloutsos, M., & Krishnamurthy, S. (2009). Lord of the links: a framework for discovering missing links in the Internet topology. *Transactions on Networking (ToN).*

Hofmann, H., Kafadar, K., & Wickham, H. (2011). *Letter-value plots: Boxplots for large data* (Tech. Rep.). had.co.nz.

How multi-cloud business models will shape the future. (2018). `https://www.cloudcomputing-news.net/news/2018/oct/05/how-multi-cloud-business-models-will-shape-future/`.

Huffaker, B., Fomenkov, M., & claffy, k. (2014). DRoP:DNS-based Router Positioning. *SIGCOMM CCR.*

Huffaker, B., Fomenkov, M., et al. (2014). Drop: Dns-based router positioning. *SIGCOMM CCR.*

Huffaker, B., Keys, K., Fomenkov, M., & Claffy, K. (2018). *As-to-organization dataset.* `http://www.caida.org/research/topology/as2org/`.

Hwang, F. K., & Richards, D. S. (1992). Steiner tree problems. *Networks.*

Hyun, Y. (2006). Archipelago measurement infrastructure. In *CAIDA-WIDE Workshop.*

IBM bets on a multi-cloud future. (2018). `https://www.zdnet.com/article/ibm-bets-on-a-multi-cloud-future/`.

IP2Location. (2015). *IP2Location DB9, 2015.* `http://www.ip2location.com/`.

IP2Location. (2018). *IP address geolocaiton.* `https://www.ip2location.com/database/ip2location`.

IPFS. (2019). *IPFS is the Distributed Web.* `https://ipfs.io/`.

Jacobson, V. (1989). *traceroute.* `ftp://ftp.ee.lbl.gov/traceroute.tar.gz`.

Jonathan, A., Chandra, A., & Weissman, J. (2018). Rethinking adaptability in wide-area stream processing systems. In *Hot topics in cloud computing.* USENIX.

Kang, M. S., & Gligor, V. D. (2014). Routing bottlenecks in the internet: Causes, exploits, and countermeasures. In *Computer and communications security.* ACM.

Kang, M. S., Lee, S. B., & Gligor, V. D. (2013). The crossfire attack. *Symposium on Security and Privacy*.

Katz-Bassett, E., Scott, C., Choffnes, D. R., Cunha, Í., Valancius, V., Feamster, N., ... Krishnamurthy, A. (2012). LIFEGUARD: Practical repair of persistent route failures. In *SIGCOMM.*

Keys, K., Hyun, Y., Luckie, M., & Claffy, K. (2013). Internet-Scale IPv4 Alias Resolution with MIDAR. *Transactions on Networking (ToN).*

Khan, A., Kwon, T., Kim, H.-c., & Choi, Y. (2013). AS-level topology collection through looking glass servers. In *Internet Measurement Conference (IMC).*

Klöti, R., Ager, B., Kotronis, V., Nomikos, G., & Dimitropoulos, X. (2016). A comparative look into public ixp datasets. *SIGCOMM CCR.*

Knight, S., Nguyen, H. X., Falkner, N., Bowden, R., & Roughan, M. (2011). The Internet topology zoo. *Selected Areas in Communications*.

Kotronis, V., Klöti, R., Rost, M., Georgopoulos, P., Ager, B., Schmid, S., & Dimitropoulos, X. (2016). Stitching Inter-Domain Paths over IXPs. In *Symposium on SDN Research.* ACM.

Kotronis, V., Nomikos, G., Manassakis, L., Mavrommatis, D., & Dimitropoulos, X. (2017). Shortcuts through colocation facilities. In *Internet Measurement Conference (IMC).* ACM.

Krishna, A., Cowley, S., Singh, S., & Kesterson-Townes, L. (2018). *Assembling your cloud orchestra: A field guide to multicloud management.* `https://www.ibm.com/thought-leadership/institute-business-value/report/multicloud`.

Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., & Jahanian, F. (2010). Internet inter-domain traffic. In *SIGCOMM.* ACM.

Lad, M., Oliveira, R., Zhang, B., & Zhang, L. (2007). Understanding resiliency of internet topology against prefix hijack attacks. In *International conference on dependable systems and networks.*

Lai, F., Chowdhury, M., & Madhyastha, H. V. (2018). To relay or not to relay for inter-cloud transfers? In *Workshop on hot topics in cloud computing.*

Li, L., Alderson, D., Willinger, W., & Doyle, J. (2004). A first-principles approach to understanding the internet's router-level topology. In *SIGCOMM CCR.*

Limelight. (2017). *Private global content delivery network.* `https://www.limelight.com/network/`.

Lodhi, A., Larson, N., Dhamdhere, A., Dovrolis, C., et al. (2014). Using peeringDB to understand the peering ecosystem. *SIGCOMM CCR.*

Luckie, M. (2010). Scamper: a scalable and extensible packet prober for active measurement of the internet. In *Internet Measurement Conference (IMC).*

Luckie, M., & Beverly, R. (2017). The impact of router outages on the as-level internet. In *Sigcomm.*

Luckie, M., Dhamdhere, A., Huffaker, B., Clark, D., et al. (2016). bdrmap: Inference of Borders Between IP Networks. In *Internet measurement conference (imc).*

Luckie, M., Huffaker, B., Dhamdhere, A., & Giotsas, V. (2013). AS Relationships, Customer Cones, and Validation. *IMC*. doi: 10.1145/2504730.2504735

Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., et al. (2013). As relationships, customer cones, and validation. In *Internet Measurement Conference (IMC).* ACM.

Luckie, M., et al. (2014). A second look at detecting third-party addresses in traceroute traces with the IP timestamp option. In *Passive and Active Measurement (PAM).*

Marder, A., & Smith, J. M. (2016). MAP-IT: Multipass Accurate Passive Inferences from Traceroute. In *Internet Measurement Conference (IMC).* ACM.

Mastodon. (2019). *Giving social networking back to you.* `https://joinmastodon.org/`.

Mathis, M., Semke, J., Mahdavi, J., & Ott, T. (1997). The macroscopic behavior of the TCP congestion avoidance algorithm. *SIGCOMM CCR.*

MaxMind. (2018). *GeoIP2 databases.* `https://www.maxmind.com/en/geoip2-databases`.

Megaport. (2019a). *Megaport Pricing.* `https://www.megaport.com/pricing/`.

Megaport. (2019b). *Nine Common Scenarios of multi-cloud design.* https://knowledgebase.megaport.com/megaport-cloud-router/nine-common-scenarios-for-multicloud-design/.

Microsoft. (2018a). *Azure ExpressRoute.* https://azure.microsoft.com/en-us/services/expressroute/.

Microsoft. (2018b). *ExpressRoute connectivity partners.* https://azure.microsoft.com/en-us/services/expressroute/connectivity-partners/.

Microsoft. (2018c). *Expressroute partners and peering locations.* https://docs.microsoft.com/en-us/azure/expressroute/expressroute-locations.

Microsoft. (2019). *Bandwidth Pricing.* https://azure.microsoft.com/en-us/pricing/details/bandwidth/.

Miller, R. (2015). *Regional Data Center Clusters Power Amazon's Cloud.* https://datacenterfrontier.com/regional-data-center-clusters-power-amazons-cloud/.

M-Lab. (2018). *NDT (Network Diagnostic Tool).* https://www.measurementlab.net/tests/ndt/.

Motamedi, R., Yeganeh, B., Chandrasekaran, B., Rejaie, R., Maggs, B., & Willinger, W. (2019). On Mapping the Interconnections in Today's Internet. *Transactions on Networking (ToN).*

NetAcuity. (2018). *Industry-standard geolocation.* https://www.digitalelement.com/solutions/.

Netflix. (2017a). *Internet connection speed requirements.* https://help.netflix.com/en/node/306.

Netflix. (2017b). *Open Connect Appliance Overview.* https://openconnect.netflix.com/en/appliances-overview/.

Nomikos, G., & Dimitropoulos, X. (2016). traIXroute: Detecting IXPs in traceroute paths. In *Passive and Active Measurement (PAM).*

Nomikos, G., Kotronis, V., Sermpezis, P., Gigis, P., Manassakis, L., Dietzel, C., ... Giotsas, V. (2018). O Peer, Where Art Thou?: Uncovering Remote Peering Interconnections at IXPs. In *Internet Measurement Conference (IMC).*

Nur, A. Y., & Tozal, M. E. (2018). Cross-as (x-as) internet topology mapping. *Computer Networks*.

OASIS. (2019). *MQTT.* http://mqtt.org/.

One-Way Ping (OWAMP). (2019). http://software.internet2.edu/owamp/.

Orsini, C., King, A., Giordano, D., Giotsas, V., & Dainotti, A. (2016). BGPStream: a software framework for live and historical BGP data analysis. In *Internet Measurement Conference (IMC).* ACM.

Packet Clearing House. (2017). *Routing archive.* https://www.pch.net.

Packet Clearing House. (2018). *MRT Routing Updates.* https://www.pch.net/resources/Raw_Routing_Data/.

PacketFabric. (2019). *Cloud Connectivity.* https://www.packetfabric.com/packetcor#pricing.

Padhye, J., Firoiu, V., Towsley, D., & Kurose, J. (1998). Modeling tcp throughput: A simple model and its empirical validation. *SIGCOMM CCR.*

Padmanabhan, V. N., & Subramanian, L. (2001). An investigation of geographic mapping techniques for internet hosts. In *SIGCOMM CCR.*

Palmer, C. R., Siganos, G., Faloutsos, M., Faloutsos, C., & Gibbons, P. (2001). The connectivity and fault-tolerance of the internet topology. In *Workshop on Network-Related Data Management (NRDM).*

PeeringDB. (2017). *Exchange Points List.* https://peeringdb.com/.

PeerTube. (2019). *Join PeerTube.* https://joinpeertube.org/.

Plaven, G. (2017). *Amazon keeps building data centers in umatilla, morrow counties.* http://www.eastoregonian.com/eo/local-news/20170317/amazon-keeps-building-data-centers-in-umatilla-morrow-counties.

Pureport. (2019). *Pricing - Pureport.* https://www.pureport.com/pricing/.

Quan, L., Heidemann, J., & Pradkin, Y. (2013). Trinocular: Understanding internet reliability through adaptive probing. In *SIGCOMM CCR.*

Richter, P., Smaragdakis, G., Feldmann, A., Chatzis, N., Boettger, J., & Willinger, W. (2014). Peering at peerings: On the role of IXP route servers. In *Internet Measurement Conference (IMC).*

RIPE. (2018). *Routing information service (ris).* https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

RIPE. (2019). *RIPE RIS.*

RIPE NCC. (2016). *RIPE Atlas.*

Robusto, C. C. (1957). The cosine-haversine formula. *The American Mathematical Monthly*.

SamKnows. (2018). *The internet measurement standard.* `https://www.samknows.com/`.

Sanchez, M. A., Bustamante, F. E., Krishnamurthy, B., Willinger, W., Smaragdakis, G., & Erman, J. (2014). Inter-domain traffic estimation for the outsider. In *Internet Measurement Conference (IMC).*

Sánchez, M. A., Otto, J. S., Bischof, Z. S., Choffnes, D. R., Bustamante, F. E., Krishnamurthy, B., & Willinger, W. (2013). Dasu: Pushing experiments to the Internet's edge. In *NSDI.*

Scheitle, Q., Gasser, O., Sattler, P., & Carle, G. (2017). Hloc: Hints-based geolocation leveraging multiple measurement frameworks. In *Network Traffic Measurement and Analysis Conference.*

Schlinker, B., Kim, H., Cui, T., Katz-Bassett, E., Madhyastha, H. V., Cunha, I., ... Zeng, H. (2017). Engineering egress with edge fabric: Steering oceans of content to the world. In *SIGCOMM.*

Schulman, A., & Spring, N. (2011). Pingin'in the rain. In *Internet Measurement Conference (IMC).*

Shavitt, Y., & Shir, E. (2005). Dimes: Let the internet measure itself. *SIGCOMM CCR.*

Sherwood, R., Bender, A., & Spring, N. (2008). DisCarte: A Disjunctive Internet Cartographer. In *SIGCOMM CCR.*

Siganos, G., & Faloutsos, M. (2004). Analyzing BGP policies: Methodology and tool. In *INFOCOM.*

Singla, A., Chandrasekaran, B., Godfrey, P., & Maggs, B. (2014). The internet at the speed of light. In *Proceedings of hot topics in networks.*

Sodagar, I. (2011). The MPEG-dash standard for multimedia streaming over the internet. In *IEEE MultiMedia.*

Spring, N., Dontcheva, M., Rodrig, M., & Wetherall, D. (2004). *How to resolve IP aliases* (Tech. Rep.). Univ. Michigan, UW CSE.

Spring, N., Mahajan, R., & Wetherall, D. (2002). Measuring isp topologies with rocketfuel. *SIGCOMM CCR.*

Sundaresan, S., Burnett, S., Feamster, N., & De Donato, W. (2014). BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks. In *Usenix annual technical conference.*

Sundaresan, S., Feamster, N., & Teixeira, R. (2015). Measuring the Performance of User Traffic in Home Wireless Networks. In *Passive and Active Measurement (PAM).* ACM.

Tariq, M. M. B., Dhamdhere, A., Dovrolis, C., & Ammar, M. (2005). Poisson versus periodic path probing (or, does pasta matter?). In *Internet Measurement Conference (IMC).*

TeamCymru. (2008). *IP to ASN mapping.* `https://www.team-cymru.com/IP-ASN-mapping.html`.

Tokusashi, Y., Matsutani, H., & Zilberman, N. (2018). Lake: An energy efficient, low latency, accelerated key-value store. *arXiv preprint arXiv:1805.11344*.

Torres, R., Finamore, A., Kim, J. R., Mellia, M., Munafò, M. M., & Rao, S. (2011). Dissecting video server selection strategies in the YouTube CDN. In *International conference on distributed computing systems.* IEEE.

Tozal, M. E., & Sarac, K. (2011). Palmtree: An ip alias resolution algorithm with linear probing complexity. *Computer Communications*.

Triukose, S., Wen, Z., & Rabinovich, M. (2011). Measuring a commercial content delivery network. In *World Wide Web (WWW).* ACM.

University of Oregon. (2018). *University of oregon route views project.* `http://www.routeviews.org/routeviews/`.

WikiLeaks. (2018). *Amazon Atlas.* `https://wikileaks.org/amazon-atlas/`.

Williams, M. (2016). *Amazon's central Ohio data centers now open.* `http://www.dispatch.com/content/stories/business/2016/10/18/amazon-data-centers-in-central-ohio-now-open.html`.

WireGuard. (2019). *WireGuard: fast, modern, secure VPN tunnel.*

Wohlfart, F., Chatzis, N., Dabanoglu, C., Carle, G., & Willinger, W. (2018). Leveraging interconnections for performance: the serving infrastructure of a large CDN. In *SIGCOMM.*

Xia, J., & Gao, L. (2004). On the evaluation of AS relationship inferences [Internet reachability/traffic flow applications]. In *GLOBECOM.*

Yap, K.-K., Motiwala, M., Rahe, J., Padgett, S., Holliman, M., Baldus, G., ... others (2017). Taking the edge off with espresso: Scale, reliability and programmability for global internet peering. In *SIGCOMM.*

Yeganeh, B., Durairajan, R., Rejaie, R., & Willinger, W. (2019). How cloud traffic goes hiding: A study of amazon's peering fabric. In *Internet Measurement Conference (IMC).*

Yeganeh, B., Rejaie, R., & Willinger, W. (2017). A view from the edge: A stub-as perspective of traffic localization and its implications. In *Network Traffic Measurement and Analysis Conference (TMA).*

Zarchy, D., Dhamdhere, A., Dovrolis, C., & Schapira, M. (2018). Nash-peering: A new techno-economic framework for internet interconnections. In *INFOCOM Computer Communications Workshops.*

ZDNet. (2019). *Top cloud providers 2019.* `https://tinyurl.com/y526vneg`.

Zhang, M., Zhang, C., Pai, V. S., Peterson, L. L., & Wang, R. Y. (2004). Planetseer: Internet path failure monitoring and characterization in wide-area services. In *OSDI.*