Mapping PoP-Level Connectivity of Large Content Providers

Amir Farzad Reza Rejaie

ABSTRACT

Large content providers (CPs) are responsible for a large fraction of injected traffic to the Internet. They maintain multiple data centers and they connect to different ASes to relay their contents and service traffic to the rest of the Internet. In this paper we propose a novel methodology to measure and characterize large Internet Content Providers. Basically our contribution is two-fold:

- We design a targeted CP oriented measurement methodology. This methodology includes Application Level Probing method, using name aliasing, subnet discovery and optimized tracerout probing.
- We show how to clean and aggregate the collected data to effectively characterize the target CP, finding PoP locations and explaining routing policies.

As a case study we apply this methodology on Google Maps to show the feasibility of our method. Considering the constraint view of this methodology, we discuss how it is capable of discovering more peering relations than other general large scale measurements and how it unravels the routing mechanisms of Google Maps service.

Keywords

Content Providers, PoP level topology, Active Measurement

1. INTRODUCTION

Large content providers (CP) such as Yahoo, Google or Amazon are responsible for a large fraction of injected traffic to the Internet. These providers often maintain multiple data centers at different geographical locations. Each data center could be organized into one or more ASes and each AS connects to multiple ASes at different PoPs by establishing often peering relationship.

The CP's ASes play a crucial role in providing the content for the current World Wide Web. Given this role, understanding and studying the vulnerability of the CP's network on the Internet has a great importance. These vulnerabilities might be due to malicious attacks (targeted attack), natural catastrophes (random or arbitrary attacks) and even political or economical issues that cause network disconnectivity between entities. Several previous studies have analyzed the resiliency of AS level topology [1, 2]. However their analysis is based on logical AS graph. The AS level topology does not capture multiple links between entities. On the other hand when we study natural catastrophes or economical issues on the network, the geographical location of the connections should also be considered. Thus, The geo aware PoP-level topology is the proper structure for vulnerability analysis of CP networks. To address these issues it is very useful to discover the PoP level connectivity of large Content Providers and their peering relationships with other ASes and entities.

The goal of this study is to tackle these question through active measurement. There are several challenges to answer these questions including the following: First, while the block of IP addresses associated with a CP is known through BGP, it is not clear which specific IP addresses are assigned/alive and visible to outside clients, or which sub-block is allocated to each data center or each group of servers. In a general case, even the location of data centers may not be known. Second, given an IP address at a data center in a known location, it is still not trivial to identify the location of the PoP between the CP and its neighboring AS, and more importantly to aggregate multiple routers/interfaces in a single PoP. Third, in the absence of any ground truth, it is challenging to ensure that all PoPs associated with a given data center are discovered.

In order to address these issues, we engineered a novel measurement methodology for Content Providers (CP). In contrast to ASes that are directly connected to end-users, CPs have no end users. The routing policy, naming resolution and location of data centers are specified by their role that is providing content and services to other users. Considering these characteristics, we directly probe the services from the application layer (Application Layer Probing) through a web browser agent. Using this method we can obtain live IP addresses corresponding to specific services (e.g. Maps, Search, Cloud

drive). Due to dynamic name resolution of CP servers, the visibility of this probing method can be extended by DNS resolution. We also exploit subnet discovery tools to group the obtained IP addresses, so we can optimize the measurement and reduce redundant probes. Following these steps, a distributed traceroute probing is run towards collected service specific IP addresses. This measurement methodology is applied for Google Maps services. The results show that the accuracy of this targeted CP-oriented approach is acceptable. Moreover, considering the constraint view of this methodology, it is potentially capable of discovering more peering relations than other general large-scale measurements.

The contributions of this paper are summarized as follows:

- We designed an Application Layer Probing (ALP) method to collect live IP addresses for a specific service of a CP,
- A systematic approach was used to extend the visibility of target IP set. We exploited the dynamic naming aliases that works for CPs,
- The target IP set is grouped at the network layer. This approach is different from previous studies that obliviously used /24 subnets for target probing,
- A hybrid method is used for mapping the source, intermediate hops and target servers to their physical locations. We used both decoding geo-tags inside DNS names and delay based triangulation tools.
- We show how this collected data can be used to characterize the traffic path, finding PoPs and analyzing routing mechanisms of a specific CP.

We applied our methodology on Google Maps. With this case study we show the feasibility of our approach. The following analysis and findings were examined on Google Maps case study.

- We characterized the path for the traffic that goes to Google maps servers by specifying its *main hops*. We showed that the routing tends to deliver the packet to Google AS as soon as they can.
- We identified the border IPs before the Google AS. Using a clustering technique these interfaces were grouped to 19 PoP locations. These PoP locations were cross validated with PeeringDB.
- We examined that Google uses a dynamic DNS resolution to redirect the users request to different servers. In particular we provided evidence to show the content server mapping of Google Maps is geo-aware.

The following sections are organized as follows: In Section 2 an overview of content providers and their service mechanisms is proposed. Our project is inspired by many previous work from different disciplines. We review these related work in Section 3. Section 4 summarizes the four main steps of our measurement methodology. The details of these steps are explained in Sections 5-8. In Section 9 we discuss the path characteristics of CP traffics as well as the PoP detection method. Routing policy also is analyzed in this section. Finally, we conclude our work in Section 11.

2. AN OVERVIEW OF CONTENT PROVIDERS: STRUCTURE & OPERATION

In this section, we sketch an overview of the structure and operation of a large content provider. Content providers are large service providers usually deployed in multiple data centers across different geographical regions. For example Google provides search, email, music, videos, maps, storage and other services to users [3]. Microsoft also provides online gaming streams [4] and Amazon delivers video streaming services [5].

These large CPs have their own Autonomous Systems with dedicated IP prefix ranges. In order to deliver the services and data to the rest of the Internet and users, they have connections to other AS entities. Because of the main role of these providers and the large amount of traffic they exchange, CPs make many peering relationships with several ASes. For instance, the PeeringDB database [6] shows Amazon, Microsoft and Google have around 50 to 150 public peering connections to other ASes. These connections are established on Point Of Presences (PoPs) or Internet exchange points (IXPs).

Here we describe the underlying mechanism used by content providers to serve a user request. This overview helps us to develop a proper method to probe the services and to get more potential information from inner infrastructure of these content providers.

As a case study, we focused on Google services. Users typically go to the Google website and select one of the services (Gmail, video, cloud drive, maps) using a web browser. The contents and the requested service are provided by *Google* content servers. These servers may be different from the original web servers that are initially called by the browser. During our experiments, we discovered the following sequence of operations: When a user goes to the Google website and opens a specific service, the following steps takes place as shown in Figure 1.

Each Google service content is specified by a URL. Suppose that the user is requesting a page from Google Maps. When a user goes to the Google maps website, or for instance opens any URL of the form `https://maps.google.com/maps?q=eugene` on an existing Google web

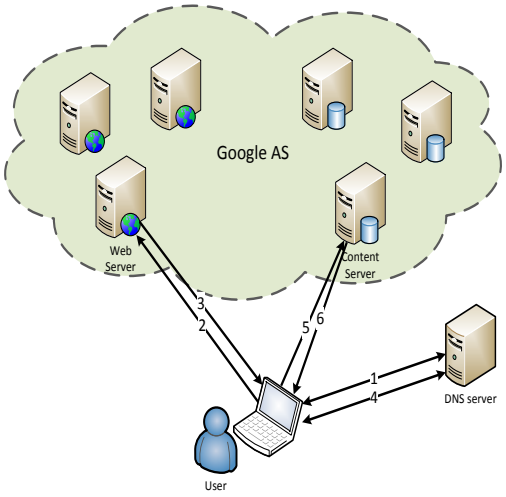

Figure 1: Sequence of steps in *Google* service delivery to user

page using his web browser, the web browser resolves the hostname maps.google.com using the local DNS server. Then the browser generates a HTTP request to the resolved server IP address. This HTTP request is directed to one of the Google web servers returned by the Google DNS system. The web server returns a HTML page with embedded images and Ajax objects, e.g., `http://ssl.gstatic.com/gb/images/`, pointing to the service provider's content server.

Then the browser tries to load those embedded elements which causes another round of DNS resolution. For example `http://ssl.gstatic.com/gb/images/` is resolved to get the images of map for a specific request. In fact, the web servers may redirect the maps request to another web server. The reason can be load balancing purposes or the availability of the data on another machine. We observed that Google uses both DNS resolution and HTTP redirection. In order to get target IP addresses and hostnames, we should consider both web servers and content servers. This information is logged in the traces of our measurement platform and provides us the raw data for analysis.

Large content providers have data centers across the world. They use various strategies to *replicate* content and services and *balance* the load between their servers. Dynamic name resolution is a popular technique for these providers [7]. Also we observed that load balancing strategies cause the name servers to resolve the one hostname to different IPs related to the data centers. As we discuss in subsection 9.3 the name resolution strategy depends on the geographical location of users. Thus, it is essential to probe the services from different locations, at different times and with different input parameters. We designed our active measurement methodology considering these facts.

3. RELATED WORK

In this project we used and bridged several techniques, tools and ideas from *Internet mapping*, *geolocation services* and recent research for *content provider* analysis. We tried to use the best practices in current literature, datasets and toolkits. Nevertheless, the limitation, validity and accuracy of these tools and ideas have been considered. Our measurement and data analysis process is combined by these ideas and related work. Our methodology demonstrates how to analyze and map the connectivity of *content providers*. We believe that our designed methodology is more than the sum of its parts and could be used as a general framework for the *PoP-level mapping of Large Content Providers*.

3.1 Internet Mapping Inference

Generally, two major types of measurements have been available for the Internet Topology Inference: i) Active measurement and ii) Passive measurement. Active measurement is usually done via active probing tools such as `ping`,`traceroute` and `mrinfo`; While passive measurement exploits the available network data, for instance BGP tables and BGP updates, to infer the Internet map [8]. The Internet topology mapping could itself be seen from different granularities and perspectives: IP interface level, router level, PoP level and Autonomous System (AS) level [9].

The project Rocketfuel [10] targets Geo-coverage of ISP topologies. Their measurement method includes active traceroute probing via a set of vantage points. It uses the BGP tables to focus the measurements and *path reduction* technique to reduce and optimize the number of probes. In the Rocketfuel project, interfaces are resolved to routers by the *ally*[11] alias resolution tool. The routers are also identified and geographically annotated using the *UnDNS* tool[11]. The *UnDNS* tool uses the naming conventions embedded in the host names of the routers to discover their geographical locations. The routers are grouped together to finally infer the PoP-level topology of an ISP.

Keys [12] has discussed the incompleteness and inaccuracy of the *ally* tool that Rocketfuel uses for alias resolution. Keys states that the common IP ID counter idea, which is used by *ally* merely works with the operating systems that monotonically increase the IP identification of the packets. Garcia *et al.*[13] and Hu *et al.*[14] show that this is not the case for many operating systems. Zhang *et al.* [15] criticize the idea of using DNS names to extract the geolocations of the routers. They mention that the location code in DNS names may be misnamed. Zhang *et al.* show that the DNS misnamings only consist of 0.5% of IP addresses in their specific evaluation data. Nevertheless, the topological impact of

these misnamings is much larger. They found that 11% of the edges for corresponding network topology are indeed false edges. Furthermore, in the subsection 9.2 we discussed that a significant portion of the IP interfaces may not have a DNS name or even the location information is not necessarily coded in the DNS names. Therefore, other methods should be considered for PoP-location discovery.

Following the Rocketfuel project, several attempts have been made to gather and infer an Internet map through active probing methods. The Ark (CAIDA) [16] project uses the large-scale traceroute-based measurement to obtain both IPv4 and IPv6 network topology. It exploits a set of distributed monitors to probe all routed /24's. Ark effectively uses the *scamper* [17] tool. The *scamper* tool supports IPv4, IPv6, traceroute, and ping. It is able to probe TCP, UDP and ICMP protocols and *paris-traceroute* [18] variations. They also provide the AS link data set derived from the IP paths of the topology dataset. Moreover, combination of multiple alias-resolution techniques is used to infer the *router-level topologies* from the data. Detected routers are mapped to a geographical location using the *MaxMind* [19] geolocation service. Part of the Ark dataset is publicly available at [20].

The Distributed Internet Measurements and Simulations (DIMES) [21] project is an active probing project that relies on a public tool that users voluntarily download and run on their machines. This tool consumes a small amount of CPU and traffic to do ping and traceroute at a low rate. The authors of the DIMES project argue that the quantity and the diversity of the monitors plays an important role in obtaining an accurate Internet map. They claim that the DIMES distributed campaign has an effective ability to gather an accurate map [22]. Their data is publicly available at [21]. The DIMES data has been used in [23, 24] to detect the PoP geolocation of specific ISPs.

iPlane [25] is a research project which has used 300 PlanetLab [26] machines to collect the internet topology. iPlane has generated an annotated map of Internet topology. The focus of this project is on core Internet backbone. They use several techniques to efficiently select the target IPs and to cluster the interfaces and annotate the routers and PoPs. In the iPlane project, target IPs are selected from the basis of BGP atoms. They also exploited the public looking glass/traceroute servers in addition to PlanetLab nodes for low intensity probing. In order to group and annotate the core Internet topology, the authors used alias resolution, location codes in DNS names and delay based clustering techniques. Their data is publicly available at [27].

The three aforementioned projects (Ark, DIMES and iPlane) have tried to get a picture (with different granularities) from the whole Internet. They used active probing tools and particularly `traceroute`. The classical traceroute has been found to have major problems, in particular due to load balancing mechanisms on intermediate routers [28]. Augustin *et al.* introduced the *Paris traceroute* [18] that is able to avoid or prevent some of these problems [29, 30]. The *Paris traceroute* techniques however, have not widely adopted. Among the three aforementioned projects, Ark (CAIDA) is the only large scale project that uses the *Paris traceroute* as a part of the *scamper* tool.

Beverly *et al.*[31] propose and analyze some techniques to shorten the Internet mapping cycle time. They investigate the Ark and iPlane datasets as well as their own active probing to find and select a proper target in a subnet and to reduce the redundant probes. Tian *et al.*[32] also exploit similar heuristics to reduce the redundant probes and optimize the mapping process. They use these techniques to map the China's Internet. As we describe in Table 4 we implement a mixture of different techniques to select the target IPs to probe. Our approach is different from aforesaid methods that use the BGP tables to aggregate the address space.

Chen *et al.* [33] exploit BitTorrent clients in an opportunistic approach to run P2P traceroutes. They discuss that using this method they are able to use a large number of monitors which reveals much more AS relationships in the inferred topology.

The `mrinfo` tool was used in MERLIN project to obtain the router-level topology of a targeted ISP[34]. `mrinfo` uses the IGMP protocol to collects all of the IPv4 multicast interfaces of a router as well as its multicast neighbors. In a later work, the authors of MERLIN proposed a hybrid method that uses the `mrinfo` as a complementary method to `traceroute` to expand the coverage of the topology measurement [35].

The BGP routing datasets also have been considered to infer the AS level Internet map. The Oregon Route-Views project [36] and RIPE RIS project [36] provide public datasets that are obtained from BGP routing tables and updates. However, Using these datasets for AS level Internet map have technical limitations [37, 38]. Augustin *et al.* investigate the IXP databases as well as BGP dataset and traceroute data to obtain AS level map and peering relations. They discuss that focusing on IXP entities can reveal many AS relations that are missing in the BGP datasets.

There are several papers that have exploited the topology dataset to analyze, characterize and model the Internet topology. The details of these research work is beyond of the scope of our paper. The interested reader may refer to [39, 40, 9]. The Internet topology models and characteristics gained a lot of attention specifically from the *complex network* science research society. Nevertheless, controversy exists over these characteristics and models in the current literature. In particular, the

power-low degree distribution and the claims about the Internet resiliency have been highly criticized by Willinger *et al.* [41, 42].

3.2 Internet Geolocation

The Internet geolocation techniques are used to determine the physical location of users and machines. These techniques are basically used to propose location based services to end users and devices. The accuracy of these services, nevertheless are criticized by some recent studies [43, 44]. In particular, the results of these services are less accurate for the Internet routers or the IP addresses that belong to content providers. In this paper, we use geolocation techniques to discover the location of PoPs and servers of the content provides. The location codes in the DNS names of the routers can be used to infer the physical location of those machines. Despite of the fact that these method has some inaccuracies [15], not all of the routers have location code in their DNS names. Moreover, some of the routers does not have DNS names. In this case, the RTT delay from `ping` probes can be computed to estimate the geolocation of these machines. Both methods were discussed in [45].

3.3 Content Provider Analysis and Other Related Work

The analysis of content providers have recently received attention in Internet measurement community. Adhikari *et al.* analyze the YouTube [46], Netflix [47] and Hulu [48] content providers. They run active measurement from PlanetLab to uncover the architecture and service strategy of these providers. The load balancing mechanisms and the replication policy of the contents on multiple data centers are discussed in these papers. The focus of these papers, unlike the aim of our work, is not the PoP discivery and geographical maps of the content provides.

We also were inspired by miscellaneous ideas from other measurements and tools. The idea of application level probing which we explain in section 5, is similar to the measurement approach in [49]. However, our probing method is done via an automatic software and is logged in the proxy server side. The details of the subnet discovery tool that we exploited in section 7 is described in [50].

4. OUR APPROACH: AN OVERVIEW

Generally, this project is about internet measurement. We use active measurement methods to capture the IP allocation strategies as well as the connectivity structure of a CP. We effectively use the probing tools at application level and traceroute tool to capture the paths that packets traverse from source IP to destination. Care is needed when we use these tools. We know that traceroute has primarily designed for network administrative purposes. When we infer the geo-aware connectivity of a CP and its neighbors, we consider the limitation of the measurement and the data that we use and collect. These limitations include traceroute artifacts [30] and anomalies, limited and biased view of the PlanetLab vantage points. Interested reader may refer to [38].

We use traceroute to find PoPs and connectivity of a target CP. We believe this active measurement tool is sound for our purpose. As we discuss in subsection 9.3 almost there is no path diversity for entering a particular CP. Thus, regardless of the bias of the routes on the Internet graph, or even the artifacts of the traceroutes on the intermediate IPs, the routers on PoPs are fixed in traces. We find and group these IP addresses to discover PoP locations of the target CP.

The Approach that we exploited in this project targets a particular CP at a time. Evidences from previous Internet measurement studies [51] show that targeting a specific network increases the accuracy and completeness of the collected data. Contrary to other Internet measurement studies like iPlane [25] and DIMES [21], we focus on CPs network.

In order to probe Content Provider's AS, like any other AS, we need live and responsive IP addresses inside the target network. However, CPs are different from regular ISPs in that they do not have user and end host machines. Therefore, unlike the measurements in [33] and [52] We cannot find live IP address by P2P IP users. Moreover, we stated in section 5 there might be large gaps in IP space of the CPs so in this case we cannot obliviously use the common /24 subnet probing method [31]. To solve this issue, we designed an Application Level Probing (ALP) method to obtain live IP addresses by probing CP's services.

Our methodology is engineered specifically for CP services and consists of 4 major steps:
1) Application Level Probing: We probe the target CP by requesting its content or services in order to discover its valid IP addresses. These probing is conducted from geographically distributed vantage points. Thus we are albe to track the strategy that the CP uses to map requests from different geographical or network locations to different data centers. Details of our methodology for application level probing along with the development of related measurement tools are explained in Section 5.
2) Using name aliasing to find more IP addresses: Given the discovered DNS names through the application level probing, we lunch DNS resolution from open DNS servers to discover more IP addresses. This method is complementary to application level probing for finding target IP addresses. Further details of this step is described in Section 6.
3) Subnet Discovery: We used the Xnet [50] subnet dis-

Table 1: Summary of the methodology

Method	Description	Goal	Section
application level probing	using an automated tool on top of a browser to probe the content provider services	obtain the IPs of the responding service and identify service characteristics	section 5
using open DNS Server to Get More IPs	to extract the domain name patterns of the CP servers and use the DNS servers to discover more servers/IPs	to extend the service domain names and IP addresses	section 6
subnet Discovery	using the *Xnet* tool to discover the layer 2 subnet of the target CP	grouping the IPs to subnets to reduce the redundant traceroute probes and group the services based on IP prefix	section 7
distributed traceroute Measurement	to probe the CP's IP addresses from a set of distributed VPs	to extract the interface level path and connectivity map	section 8

covery tool to group the IP addresses -that we found from previous steps- to layer 2 clusters. Since each layer 2 subnet is in the same physical Ethernet network the traceroute measurement could be done on one IP on each cluster so we can avoid redundant traceroutes .Details are discussed in Section 7.

4) Conducting Traceroute Measurement: We conduct traceroute measurement from distributed vantage points (*e.g.*, PlanetLab nodes) towards the discovered IP groups. To reduce the number of the required measurement, we leverage the characteristics of the discovered IP address in the previous part. Further details of this step is described in Section 8.

The description and goal of each step is summarized in Table 1. Details of each step is discussed in sections 5 to 8.

The relation of each part and the workflow of our measurement methodology is depicted in Figure 2. Using the IP addresses obtained from ALP and open DNS servers, we conduct traceroutes towards target CP. Furthermore we specify the AS path and main hops from traceroute measurement for further analysis. The location of PoPs and other higher level mechanisms (e.g. routing mechanism in CP) can be inferred from this data.

Our analysis are performed for a specific service of a target CP. Here, we primarily focus on Google as our target CP. However, we can repeat this analysis for different services of a single CP to determine how these services are provided across different data centers. We suspect that the main change in our measurement methodology for different CP is the details of application level probing step.

5. APPLICATION LEVEL PROBING (ALP)

In order to perform active measurement we need responsive IP addresses inside the CP network. We designed a new toolkit to find these IP addresses. The idea is using a browser agent at the application layer to mimic the natural way that a user demands for a service. The user requests are served from servers with live IP addresses. These servers can be probed automatically with the same method in the browser software.

We probe the content provider services via an automated tool on top of the Firefox v14.0.1 browser. The tool is an automated script, based on the iMacro v7.6 add-on [53]. Also the firebug v1.10 add-on is activated on Firefox to log the network traffic of these services. We can distribute this toolkit distribute this toolkit on PlanetLab machines so the probing process is done simultaneously on different geographical locations. However PlanetLab vantage points are not capable of running a fully functional version of Firefox. We solve this issue by running the browser on local machines in our lab and then redirect the traffic to proxy servers. these proxy servers are on PlanetLab vantage points. In this way the target CP can be probed through geographically distributed VPs. Figure 3 shows this architecture.

Our ALP platform was run on 5 virtual machines at university of Oregon with dedicated valid IP addresses. We did the measurement in a period of three and a half week. We configured The ALP browsers to use the PlanetLab nodes as proxy servers. Thus for the Google Maps servers, the probes are done through different IP addresses in different locations. Totally, we exploited 376 PlanetLab vantage points in 35 countries. The Squid proxy server v 2.6 was logging the details of the IP and TCP connections on each vantage point while we did the measurement. The ALP agents were configured to use a list of city names of the capital of the countries [54] for keyword search on the Google Maps. This list for different geographical locations can potentially show the dependence of the service to the location of the probing. Changing the proxy servers and keyword in search was done with round robin order from the available lists. On each VP every 3 min the iMacro extension changes the keyword and proxy from the cor-

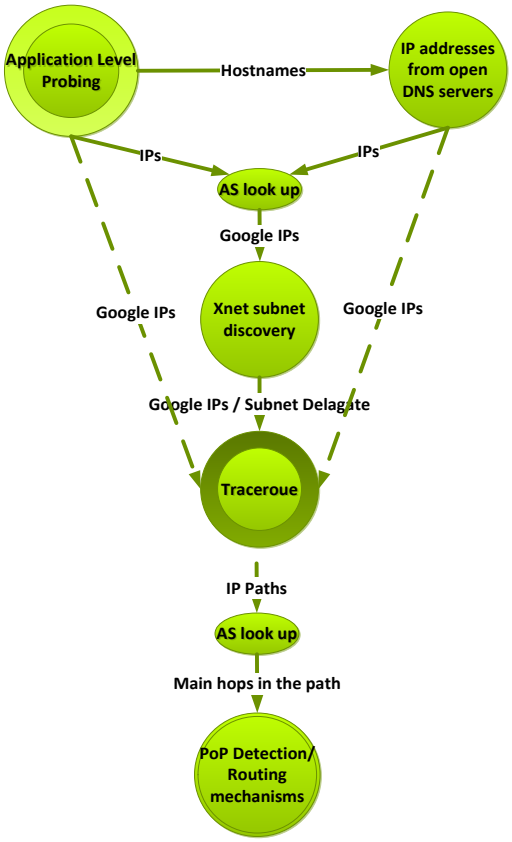

Figure 2: measurement methodology workflow

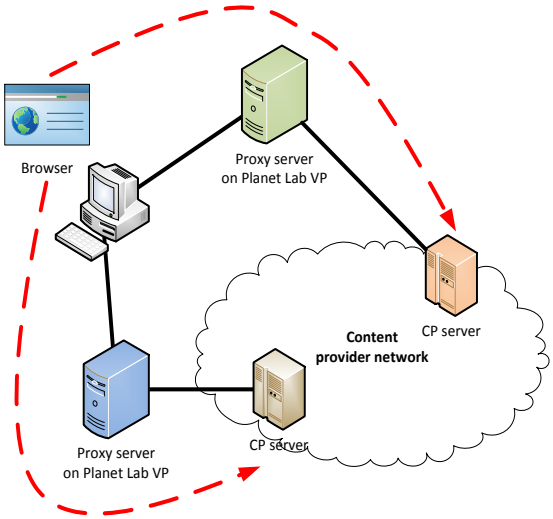

Figure 3: Application Level Probing (ALP) architecture

responding lists and did the probe automatically.

We observed that corresponding HTML pages are resolved by different server connections. For example the images of the map itself come from server A, the corresponding Ads come from server B, and the logo and toolbar images come from server C. Therefore, for each probe the proxy servers logged several servers and TCP connections. During the three and half week of lunching this platform, we did 44,600 Google Maps probes with different combinations of keyword/VP.

Overall, we collected more than 175,000 VP/Google IP pairs. Note that these pairs were not unique since the requests might be resolved by the same server.

Although, the probing process on browser is round-robin, we observed that the number of probes for VPs are not uniform. Figure 4 shows the cumulative distribution of the number of probes for VPs. Our investigation shows the faulty nature of PlanetLab affects this probing method, causes this difference on number of probes. During our analysis, we considered this bias. In order to have more reliable data, we eliminated the log files of VPs with less than 100 probes. The results of this report is extracted from the whole ALP measurement platform and except for the cases that we explicitly mention, this bias for individual VPs does not affect the correctness of our results.

The speed of gathering unique IPs is an important factor of measurement performance. Figure 5 shows the rate of new discovered IP address per probe for the whole distributed ALP platform. The average speed of discovering new IP addresses is about 11 IP per 1000 probe. The fitted curve shows, there is no sign of saturation for the found IPs. As we discussed in Section 4 ALP is a sound method to start gathering the CP IP addresses. However, Figure 5 shows that we need an alternative method. We exploited open DNS servers to discover more IP address in CP network.

6. USING NAME ALIASING TO GET MORE IPS

As we described in Section 2, when a user request a service from a CP (e.g. Google Maps), in the first step, a web page is sent to user, includes DNS names to content servers. For instance DNS name mts1.google.com is for map images. A round of dynamic DNS resolution is done for mts1.google.com and the traffic is redirected to corresponding Google data center. The relation of mts1.google.com to Google server is 1-to-many and the mapping is dynamic. In subsection 9.3 We discuss that this resolution is geo-aware. In this section we use the idea that resolving a Google DNS name from various DNS servers, returns different server addresses, so we can take advantage of this fact to collect more CP IP addresses by DNS lookup.

The DNS names for each service are taken from ALP

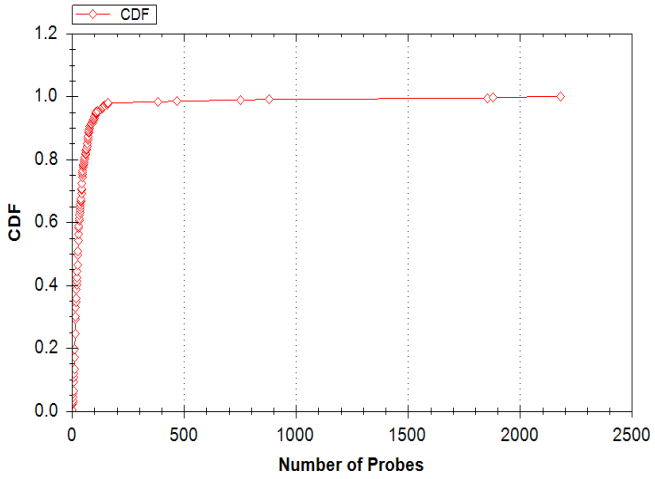

Figure 4: CDF of the number of squid probes on each PlanetLab VP for ALP.

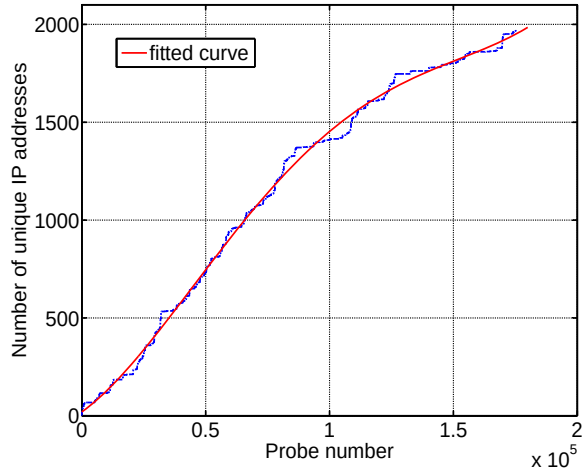

Figure 5: Number of unique IP addresses discovered from Google by application level probing. As the fitted curve shows, there is no sign of saturation for the found IPs.

log files. Figure 6 shows the DNS name servers for Google maps and the percentage of resolution for each name during ALP measurement. In these names, the symbol "#" is a digit number between [0-9]. The percentage of each name, shows the role of the DNS alias for the service. For example the number of resolution of mts#.google.com is six times more than the number of maps.google.com. Thus on average each map request is also served by six mts# server from Google.

note that, The percentage of the DNS resolution in Figure 6 are obtained from the whole aggregated ALP

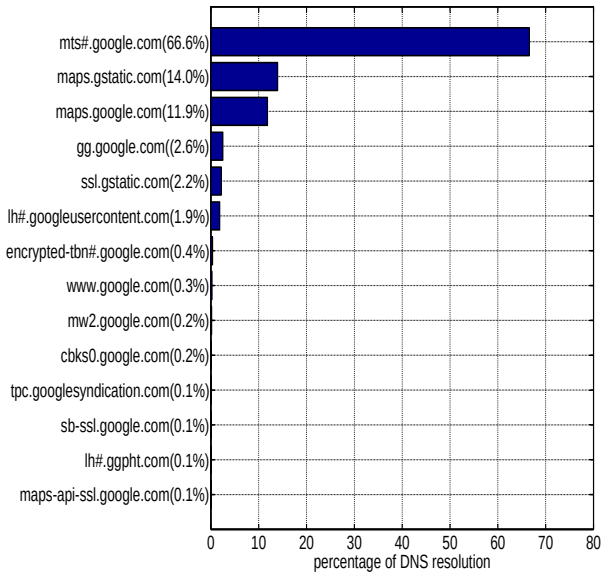

Figure 6: percentage of the DNS resolution in our ALP for Google map pages for different search keywords

data. However, The diversity of the visible DNS names from individual VPs is not uniform. Figure 7 represents the CDF of the DNS Patterns visible from each VP. The CDF shows most of the VPs observe 6-11 DNS domains. This result might be affected by the bias of the number of probes per VP. However, excluding the VPs with less than 100 probes, we believe it represents the overall diversity of the naming alias visibility on PlanetLab nodes.

Unlike the rate of new discovered IP addresses during ALP process, unique DNS patterns are saturated at the early stage of the measurement. Figure 8 shows a steep rise on the number of discovered DNS names at the beginning of the ALP measurement and then the set is saturated. This set can be used in DNS lookups to collect more IP addresses from CP. We stated that the DNS lookup returns different IP addresses from different location. We distributed this name alias lookup on more than 2000 open DNS servers [55] in 175 countries. The measurement was done over two weeks. Figure 9 shows the number of unique IP addresses discovered by open DNS servers during probes. The tail of fitted curve shows that it is saturated. According to this figure, the average speed of discovering new IP addresses is approximately 115 IPs per 1000 probe which is more than 10 times faster than ALP probing. The total unique IP addresses discovered from this phase are 3739 and they include all of the IP addresses of the ALP phase. How-

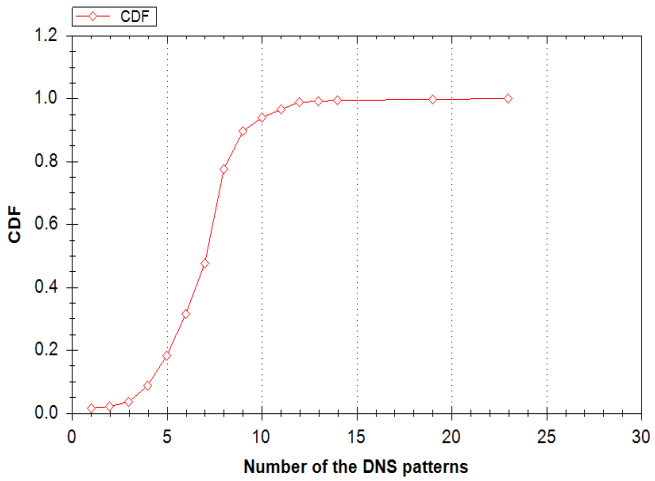

Figure 7: CDF of the DNS Patterns visible from each VP

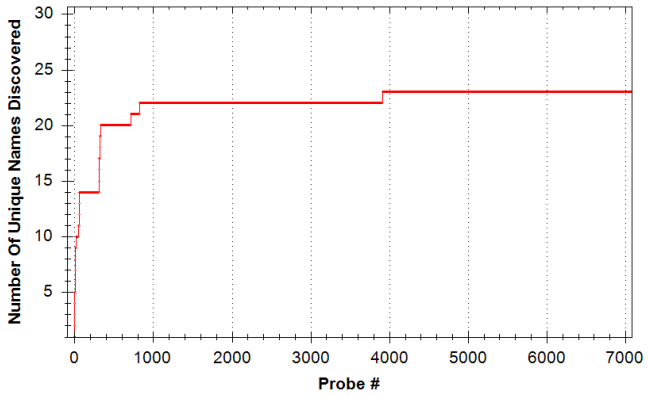

Figure 8: Number of unique DNS names discovered by ALP

ever, we could not omit the ALP technique since Figure 4b shows that in some situations during ALP we might find new DNS names that we were not able to find with less application level probes.

In the next section we discuss how subnet discovery technique can be used to group the collected IP address into layer 2 subnets. Using this technique we can reduce the redundant traceroutes for the IPs in the same subnet.

7. SUBNET DISCOVERY

A typical technique in active Internet measurement studies is to group the target IP space to /24 subnets and probe one or few IPs for each subnet [31]. On the other hand the group of subnets is one of the characteristics of a CP that represents IP allocation strategies. Our observation shows that /24 size is coarse for CPs. Also the subnet size is variable so a smaller fixed size may not help in this case. To overcome these issues we used the Xnet [50] tool that finds the network layer 2 subnets.

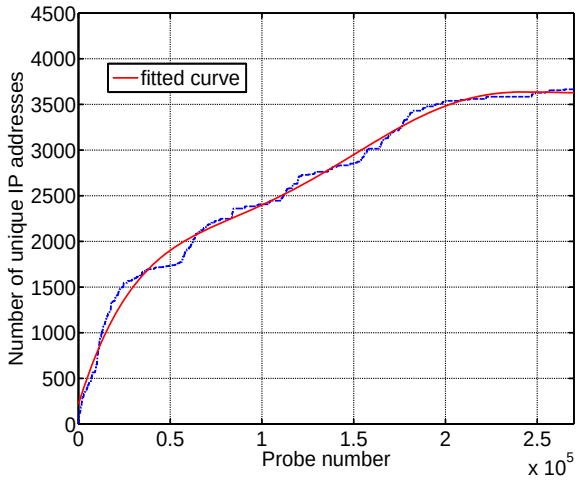

Figure 9: Number of unique IP addresses discovered by open DNS server during probes

More than 3700 IP addresses were collected by ALP and name aliasing steps. These IP addresses can be used as target IPs to do the traceroute and gather topology information as well as routing interpretation. However, some of these IP addresses are actually in the same IP prefix or physical subnets. Probing those IP addresses does not get new information. We grouped the target IP addresses based on the layer 2 physical subnets which have the same routing characteristics at the IP layer. After grouping the IP addresses, we aggregated those groups that have overlaps. These groups do not necessarily represent the complete subnet space. Because the Xnet tool method is conservative and detects the IP addresses only for live IPs and does not automatically extend the IP range. Overall we grouped about 1700 subnets from the output of Xnet tool. The subnets then merged by overlapping IP addresses in any group. In other words we made a new set by unifying of any two sets that have one ore more common elements. The result consists 129 subnet groups.

The majority of the discovered subnets include 31 IP addresses or less. In Figure 10 we plot the cumulative distribution of the number of IPs per subnet. The CDF shows that more than 80% of the subnets have 31 live IP addresses corresponding to /27 Ethernet subnets.

In the sections 5 to 7 we explained the method that we use to collect and group the IP addresses we need for CP to probe for tracerout measurement. In the next sec-

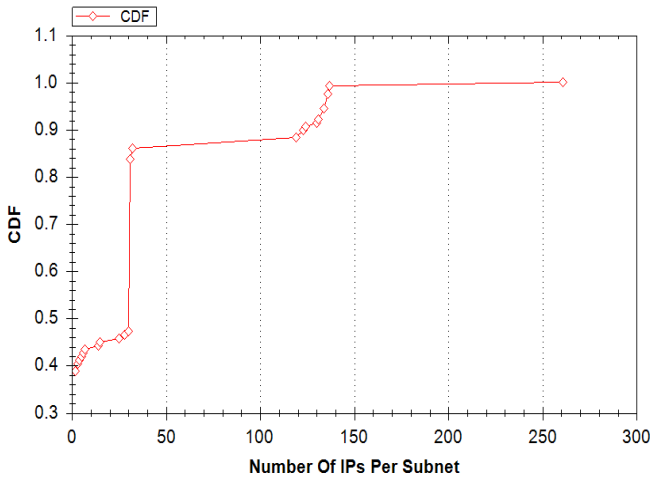

Figure 10: CDF of number of IPs per subnet

tions we explain how we effectively use these IP groups to lunch traceroute measurements so we can detect the PoPs from the traces.

8. DISTRIBUTED TRACEROUTE MEASUREMENT

One of the main goals of this project is to find the PoP level connectivity of the target content provider. We are also interested in finding the physical location of these connections. Having the target IP addresses from previous measurement steps, our approach is do active measurement with traceroute. In order to have an acceptable level of coverage, we should launch the traceroutes across multiple locations. The output of these traces helps us to discover the entry points to CP, its neighbors and possibly the location of the connection.

We distributed and lunched the traceroute measurement on 437 PlanetLab nodes. Our active probing includes a central scheduler and monitor program that distributes the target IPs on PlanetLab machines and runs traceroutes from those machines towards the target IP addresses. This distribution is essential since we want to probe the target CP from different networks across different physical locations.

We lunched a new round of traceroute every 3 hours. Each round consists of one traceroute from VP to all destinations inside CP from the set of CP IP list. The speed of one round of traceroute on VPs was not uniform. It was related to the performance of the PlanetLab machine and network speed. According to our measurements the time of one round of traceroute probing takes about 30 minutes to 2.5 hours. We set the window to 3 hours to make sure all VPs can finish the probing process within this period.

Overall 738,093 traceroutes were conducted during 8 days. The traceroute logs may have artifacts and need cleaning for our purpose. For some traces we may have missing hops (with * mark in traceroute output) or even the probe may not reach the destination. Because we need an accurate value for path length and delay, we excluded all the traceroutes even with one * in their path. By this filtering, about 35% traces were removed. There were also some other types of errors in traces like "destination unreachable" that consisted of less than 1% of the traces. Eliminating these errors we had 476,904 remaining traces. The cumulative distribution of eliminated traces per VP is plotted in Figure 11. The minimum number of eliminated traces is 200. For about 60% of the VPs, we have 500 or less removed traces. That means we have at least 1150 remaining tracerouts for these VPs. For some rare cases all of the traces has been removed.

In addition to traceroute probing, the IP for each hop was also given to reverse DNS lookup service on each VP. The VPs used their local DNS resolver to return the corresponding DNS name. In Section 3 we mentioned that the mapping between the DNS names like "mts0.google.com" to IP addresses is not 1-1 and it is dynamic. Actually the 14 patterns in Figure 6 are mapped to more than 1600 IPs. However, in the reverse DNS lookup we observed a different situation: For a specific IP, regardless of the VP, we always found a same name ending with ".1e100.net". That means for each IP in Google AS, there are two assigned names. One is dynamic for service URLs (e.g. mts0.google.com) and one static unique name (e.g. arn06s01-in-f15.1e100.net). The static names help us to identify the geo-location of Google IP addresses. More details are discussed in subsection 9.1.

9. ANALYSIS

In this section we discuss some analysis for the path characterization, PoP detection and routing mechanisms for the Google CP that are involved in the Maps service.

9.1 AS Level Path and Main Hops

We analyze the path characterization by *main hops* along the path from source VP to destination server in CP. We define main hops as six specific IP addresses as follows:

- VP: the source hop. The physical location of these nodes are known from PlanetLab database.
- LNGIP: Last Non Google IP. This is the IP just before the Google AS.
- PNGIP: Previous Non Google IP. This is the hop before the LNGIP in the traceroute path.

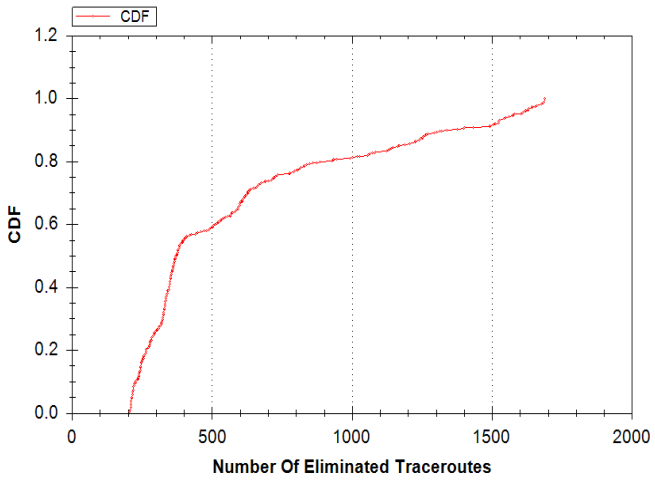

Figure 11: CDF of the number of eliminated traceroute probes per Vp

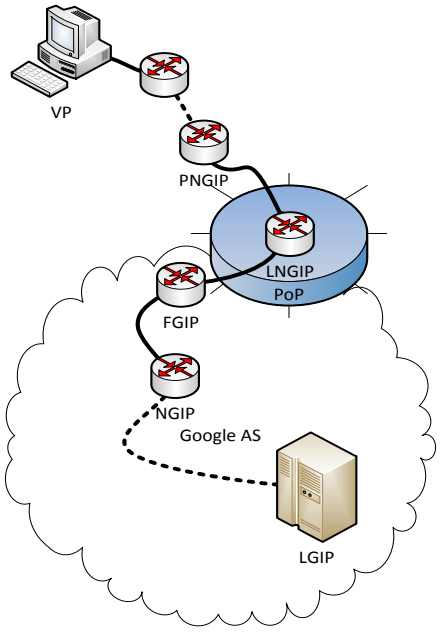

Figure 12: The role of *main hops* in the traceroute path.

- FGIP: First Google IP. First hop inside the Google AS
- NGIP: Next Google IP. The hop after FGIP.
- LGIP: Last Google IP. The destination hop.

Using these hops, we characterize the path and find the peering relationship between Google AS (or generally CP AS) with its neighbors. Figure 12 represents the position of main hops along the traceroute path.

The first step for detecting main hops is to resolve the AS of the IPs, so we can separate the Google IPs and non Google IPs. We used the *cymru*[56] IP to AS service. After detecting the LNGIP, annotating the PNGIP, FGIP and NGIP are straight forward and can be done by looking to direct LNGIP neighbors.

For path characterization and verification we measured the AS hop count, IP hop count and minimum RTT between the source and all other main hops. The cumulative distribution of these measures are plotted in Figure 13-15. They are the statistics for all the main hops for all the traceroute logs after cleaning.

The CDF of AS hop count is as expected. The hop count for FGIP, NGIP and LGIP are equal since they are in the same Google AS. In some cases the VP is directly connected to Google and then we do not have six separate main hops. For example the PNGIP and LNGIP is not defined in this case. Sometimes FGIP and LNGIP are overlapping. This means the first Google IP is actually the last one. This is the reason for the tail of the LGIP in Figure 13 which means we merged some of the six main hops.

The IP hop count distribution of PNGIP, LNGIP, FGIP and NGIP in Figure 14 is similar with one offset correspondingly. Nevertheless, the distribution of hop count for LGIP is different and shows more diversity. It shows that in some cases the destination server of Google CP is close to the entry point and in some other cases we need to pass 30 hops to reach to destination.

In Figure 15 we considered the delays of 900ms or greater as outliers and removed them. In this figure the shape of the distribution of PNGIP, LNGIP, FGIP and NGIP are also similar. The diversity and distribution of RTT for LGIP is also consistent with the diversity on hop count in Figure 14.

Figure 16 shows the delay of the packets before Google AS and the relative delay inside Google. Each point represents one relative delay measure. It is interesting that for packets the delay inside Google is more than the delay before reaching the network. It shows that routing tends to deliver the packet to Google AS as soon as they can. Then the rest of the routing is done inside Google. This result is also consistent with number of hops and RTT which shows the values are larger inside the Google network.

9.2 PoP Detection

The main "hops" from the previous section help us to detect the interfaces of the Google neighbors and detecting the PoPs. Those interfaces are actually LNGIP addresses. We use the tool UnDNS[10] to map the DNS names of those IPs to their geographical locations. However not all of the LNGIPs have DNS names. Moreover in some cases they have DNS names but they do

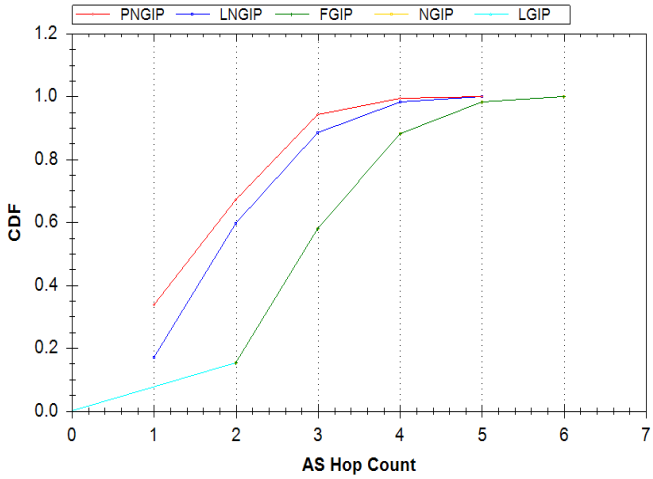

Figure 13: CDF of the AS hop count.

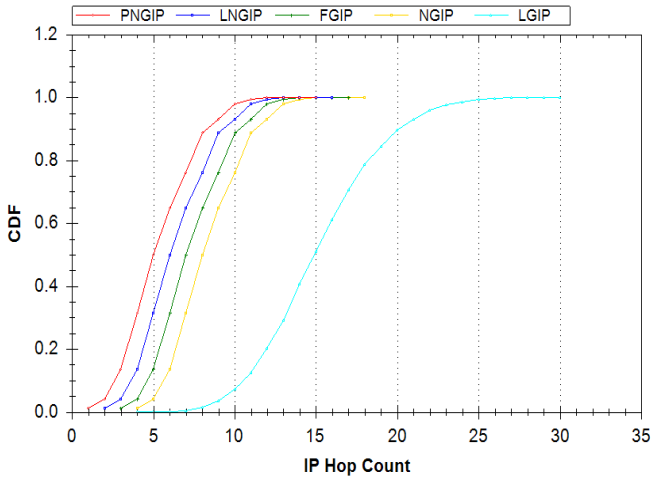

Figure 14: CDF of the IP hop count.

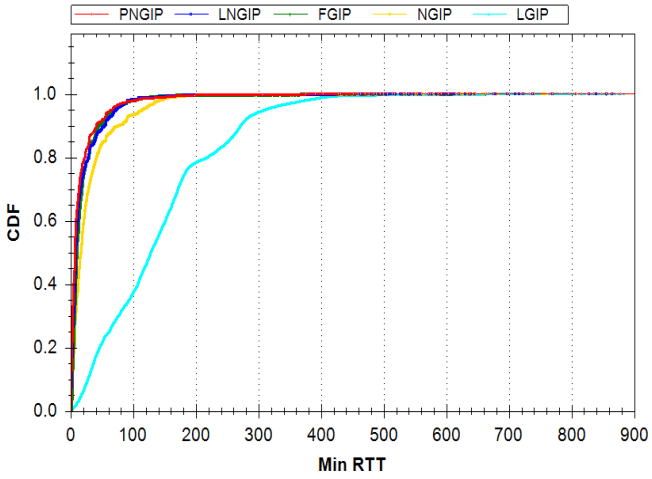

Figure 15: CDF of Min RTT.

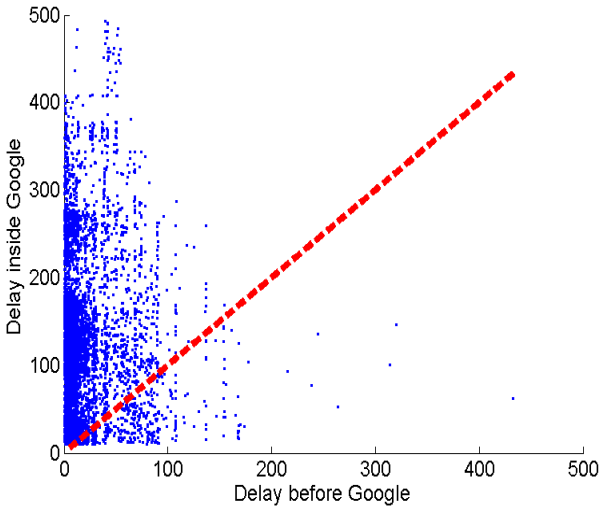

Figure 16: Scatter plot of delay measurement before vs. inside Google. The red line shows the straight line $y = x$.

not contain the geographical location or the UnDNS tool is not able to decode it. In these cases we may rely on those interfaces so that we can get the location for them and ignore the rest. Therefore the result will not be complete. But our analysis will be based on more reliable inputs.

The number of unique LNGIPs in our data is 196. 146 of them have DNS names and UnDNS can map 85 of these 146 names to geographical locations. We also found 20 IXP connections by resolving the IPs to their ASes. These IXP IPs do not have DNS names. However, we can get their physical location from their corresponding website. In this way we can resolve 20 more IP addresses.

Our PoP detection method is based on a simple location clustering approach. Here by clustering we mean clustering the interfaces in the same city to PoPs. We discuss in most cases there is only one peering AS to Google in each PoP and at most we found 3 peering in one physical location. This implies most likely the PoP locations show the actual PoP in the same facility. So in this study we consider the *physical locations* of the PoPs as a group which represents enough information for our analysis.

Adding the IPs from the IXP connection we discovered 19 PoP locations. The map of the detected PoPs is depicted in Figure 17.

Moreover Table 2 shows the list of the corresponding

Figure 17: Map of the locations of the PoP level connectivity of Google AS.

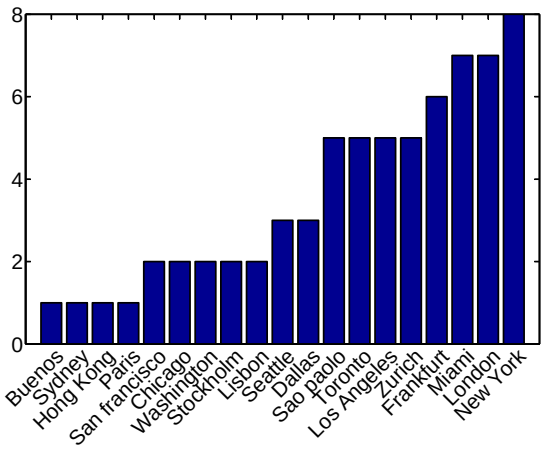

Figure 18: Number of corresponding Planet Lab sites for each PoP location (in increasing order).

PoP locations and the peering ASes with Google that we discovered. According to our results, there are two PoP locations which have three AS peering with Google (Miami and New York). Six locations has two peering and the other 11 locations have only one peering with Google.

We also examined the relation between the vantage points and the PoPs. The noticeable result is that the vantage points always relay the traffic to Google through the same PoP. We observed one exception for a vantage point in Romania belongs to AS2614 that connected to Google from three different PoPs in London, Stockholm and Frankfurt. On the other side the relation between the PoP and the PlanetLab sites varies from 1 to 8 PlanetLab site per PoP. In other words PoPs may pass traffic for one vantage pointin one PlanetLab site or in some cases 8 PlanetLab sites. Note that each site PlanetLab site may contain one or more VPs. Figure 18 shows the number of vantage points corresponds to each PoP location. We also provided the list of corresponding PlanetLab sites for each PoP location in appendix Table 3.

9.2.1 Cross Validation with PeeringDB

PeeringDB is a database to exchange information related to Peerings between ASes and IXPs. Peering DB might not be correct, updated or complete. It only shows that the connection is present at the city. It does not show the inner connection and PoP architecture. However we can cross validate the PoPs and peering relationships that we found from our methodology with data available at PeeringDB website[6].

The PeeringDB record for the main AS of Google, which is AS#15169, shows more than 140 peerings with other ASes and IXPs. These PoPs are in 54 cities. The city level location that we had discovered contains 34 Peering relations and 19 locations and all of them are available on PeeringDB. Note that in this project, we limited our study solely to Google Maps services. We suspect that in expanding the ALP measurement to other services, we will get a decent coverage of peering and PoP locations. It is worth mentioning that with our targeted method we found 18 new peerings for Google (among 34) which are not available in the large RIP BGP data set[57]. This fact shows the effectiveness of targeted active measurement.

Note that in this study the goal is to show the feasibility of the approach. Extending the measurement to other services and using more distributed vantage points can enhance the coverage of the PoPs that we discover. The PoPs we discovered are the PoPs that are visible from vantage points that we used for probing. In Figure 19 we plot the distribution of the distance between VPs and detected PoPs and also the VPs and all other PoPs -which we did not detect- from peering DB. This figure shows that the detected PoPs are actually the PoPs that are much closer to our VPs. Thus using more distributed vantage points (e.g. looking glasses) can enhance the coverage of the method. Actually the number of VPs that are included in the sites in figure Figure 18 is 137 (located in 68 unique sites) while we had used 437 PlanetLab nodes for traceroute probing. That means in these results we inevitably eliminated 300 VPs because of incomplete traceroutes or lack of geolocation resolution for corresponding PoPs.

9.3 Routing Policy

We described that the URLs like maps.google.com are resolved dynamically, based on the IP address of the user's machine. In order to find any location-aware relation for this kind of server to user association one needs to know the physical location of the user and the server machine. In our measurement the location of the

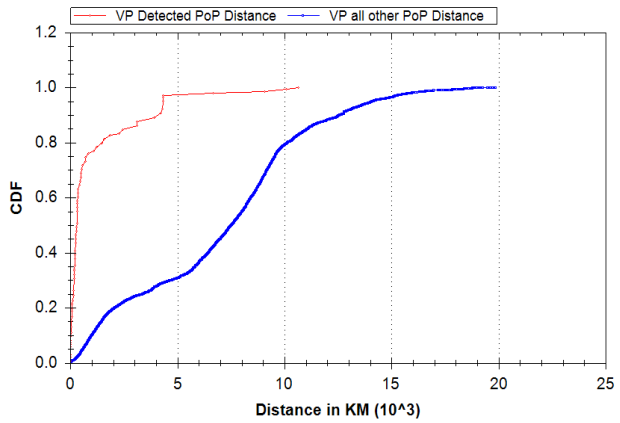

Figure 19: CDF of the distance between VP and PoPs.

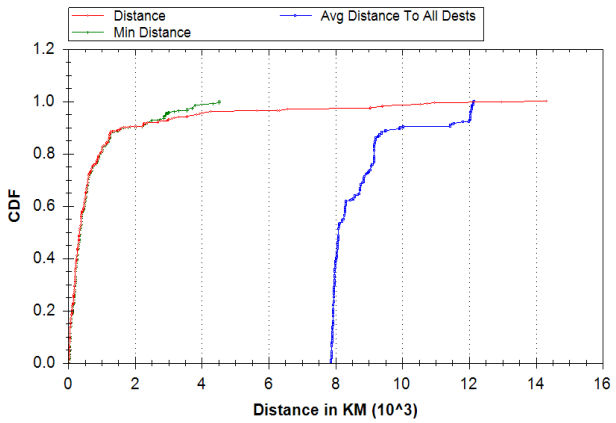

Figure 20: CDF of the distance between VP and Google destination IP.

user (vantage points) is known through the PlanetLab information center. On the other side, the location of the server machines in data centers of Google can be extracted using their DNS names.

In Section 8 it was stated that in addition to dynamic DNS names, each IP inside the Google AS has a unique alias name ending with .1e100.net. We observed two kinds of these names:

- XYZ##s##-in-f#.1e100.net which "XYZ" is a 3 letter airport code.
- XX-in-f###.1e100.net; We do not have geo tagging inside this kind of name so we should use another technique to get the location. In this case, the delay based *triangulation* [58] can help us to estimate the physical location of the server.

In both name patterns # is a one digit number. The corresponding DNS name of collected IP addresses can be grouped into similar names which have similar physical locations. For example the names "arn06s01-in-f0.1e100.net" and "arn06s02-in-f17.1e100.net" have the same airport code "arn" (Stockholm-Arlanda). Grouping all the collected 1681 server addresses inside Google network, we retrieved 58 groups. For each group, a random IP inside the set was selected for triangulation estimation. We used the Stanford TULIP [59] service for location estimation. For some of the groups, the triangulation probe did not answer. This was because at the time that we probed these addresses they were not alive.

We got the location of 15 groups from both airport decoding and triangulation. The maximum distance between corresponding locations was 721.6km and the average was 278km which shows triangulation could be an acceptable estimate for our purpose. In 13 groups we were only able to extract their airport codes and in 22 groups from second pattern we only had triangulation estimation. For the 8 remaining groups the airport decode and triangulation did not work. They were excluded from our analysis. A summary of these Google server grouping and locations is described in Table 4 in Appendix.

Figure 20 represents the CDF of the distance between VP and Google destination IP. The distance is the one that traffic selects during routing with normal name resolution on the browser. In other words, this distance is between the pairs from ALP probing. In the figure, the distribution for average and minimum distance between VP and all other destination IPs is also plotted as the base line. The distance is very close to minimum distance and had an offset of about 8000km to average distance. Clearly the routing policy of Google and data center mapping for services is geo-aware.

We also examined the diversity of entering points to Google per VP. Results showed that in more than 97% of the cases, the traffic enters through same city. The diversity for entering point for the other 3% is at maximum 3 cities. Therefore the traffic almost always enters the Google AS from the same city/location.

10. DISCUSSION

In this section we discuss the limitations of our methodology and the scope that we applied the method in our measurement case study as follows.

The main reason that we focused on Google maps in this study was to show the feasibility of our methodology. We found 19 PoPs with this method and confirmed the findings by PeeringDB data. However it is also mentioned PeeringDB shows more than 140 peering relationships between Google and other entities. Hence, we need to extend our methodology to other services to

increase the coverage while keeping the accuracy of the PoP detection method.

Currently our approach works only for services that a user can probe them through a web interface. Thus those servers that do not directly serve the web users cannot be probed by ALP method. However the subnet discovery which we explained in section 7 may cover all the alive interfaces on the same subnet and hence it is able to collect all the live IP addresses limited to the discovered subnets, regardless of the service that we probe.

In Section 9.2.1 we stated that the actual number of VPs that are included in PoP detection method is 137 while we had used 437 PlanetLab nodes for traceroute probing. We inevitably eliminated 300 VPs because of incomplete traceroutes or lack of geolocation resolution for corresponding PoPs. We suspect regardless of the probed service the VPs reach to Google through the same PoP and the PoP is the one is rather closed to VP. Therefore using more distributed VPs can enhance the coverage of our results. One idea might be using looking glasses (LG) from different entities and lunch traceroutes through them.

In addition to VP diversity, one of the other aspects that affects the coverage of our data was the traceroute artifacts. The common traceroute artifacts and anomalies were discussed in [30]. In section 8 we explained that we removed all of the suspected traces. This might reduce the coverage of the PoPs that we potentially can discover through measurements but keeps the result accurate. However the border routers before the CP maybe valid even incomplete traces. So we may double check and consider those traces as well in future analysis.

The coverage of ALP could be extended by probing more services on the CP. However, we need to repeat the ALP process on other services which are not necessarily same as Google maps. For example each service has different web interface and probably needs different keywords to search. For some specific servers like Google Drive (Cloud storage) or Google Play (provides Android Apps for portable services) we need to design specific methods for ALP.

The accuracy and the coverage of our PoP detection method depends on the geo tagging of the interfaces that we separated them as border routers. We believe the location resolution is accurate and the issue is the coverage of location resolution. In this paper we used geolocation codes embedded in the DNS names and the information from IXPs websites. We mentioned that we was not able to resolve the geolocation of a portion of these interfaces and so we excluded them from the PoP clustering method. Obviously resolving the location of those eliminated interfaces can increase the coverage of the PoP detection and increases the quality of the results.

11. CONCLUSION AND FUTURE WORK

In this report we proposed our methodology for measuring the content providers.The collected data was used to characterize the CP as well as its connections to other neighbors. We focused on Google Maps as a case study to show the feasibility of this method, to detect its PoPs and explain the routing mechanisms, IP allocation and naming conventions of Google data centers. We cross-validated the discovered PoPs with peeringDB.

Our future plans for this project are to extend the method to other Google services as well as other Networks (e.g. Yahoo, Microsoft and Amazon) to obtain a complete map of CPs. A comparative study of these network will be worthwhile to study. Particularly interesting for analyzation are the new cloud and mobile services on these CPs.

12. REFERENCES

[1] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4, pp. 175–308, 2006.

[3] Google data center locations. [Online]. Available: http://code.google.com/more/table/

[4] Microsoft data centers. [Online]. Available: http://www.globalfoundationservices.com/

[5] Amazon Global Infrastructure. [Online]. Available: https://aws.amazon.com/about-aws/globalinfrastructure/

[6] Peering DB. [Online]. Available: http://www.peeringdb.com/

[7] V. Adhikari, S. Jain, Y. Chen, and Z. Zhang, "Vivisecting youtube: An active measurement study," Technical report. http://www. cs. umn. edu/research/technical_ reports. php, Tech. Rep.

[8] M. Crovella and B. Krishnamurthy, *Internet measurement: infrastructure, traffic and applications*. John Wiley & Sons, Inc., 2006, ch. 5.

[9] B. Donnet and T. Friedman, "Internet topology discovery: a survey," *IEEE Communications Surveys and Tutorials*, vol. 9, no. 4, 2007.

[10] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 133–145, 2002.

[11] Rocketfuel project software and tools. [Online]. Available: http://www.cs.washington.edu/research/networking/rocketfuel/

[12] K. Keys, "Internet-scale ip alias resolution techniques," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 50–55, 2010.

[13] S. Garcia-Jimenez, E. Magaña, M. Izal, and D. Morató, "Validity of router responses for ip aliases resolution," *NETWORKING 2012*, pp. 358–369, 2012.

[14] X. Hu and Z. Mao, "Accurate real-time identification of ip prefix hijacking," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 3–17.

[15] M. Zhang, Y. Ruan, V. Pai, and J. Rexford, "How dns misnaming distorts internet topology mapping," in *Proceedings of the annual conference on USENIX'06 Annual Technical Conference*. USENIX Association, 2006, pp. 34–34.

[16] Archipelago measurment infrastructure. [Online]. Available: http://www.caida.org/projects/ark/

[17] Scamper. [Online]. Available: http://www.wand.net.nz/scamper

[18] Paris Traceroute. [Online]. Available: http://www.paris-traceroute.net/

[19] MaxMind GeoLite Databases. [Online]. Available: https://www.maxmind.com/app/geolite

[20] CAIDA Data - Overview of Datasets, Monitors, and Reports. [Online]. Available: http://www.caida.org/data/overview

[21] The DIMES project. [Online]. Available: http://www.netdimes.org

[22] Y. Shavitt and E. Shir, "Dimes: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 71–74, 2005.

[23] Y. Shavitt and N. Zilberman, "A structural approach for pop geo-location," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*. IEEE, 2010, pp. 1–6.

[24] D. Feldman, Y. Shavitt, and N. Zilberman, "A structural approach for pop geo-location," *Computer Networks*, 2011.

[25] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane: An information plane for distributed services," in *Proceedings of the 7th symposium on Operating systems design and implementation*. USENIX Association, 2006, pp. 367–380.

[26] PlanetLab. [Online]. Available: https://www.planet-lab.org/

[27] IPlane Dataset. [Online]. Available: http://iplane.cs.washington.edu/data/data.html

[28] B. Augustin, T. Friedman, and R. Teixeira, "Measuring multipath routing in the internet," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 3, pp. 830–840, 2011.

[29] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman, and R. Teixeira, "Detection, understanding, and prevention of traceroute measurement artifacts," *Computer Networks*, vol. 52, no. 5, pp. 998–1018, 2008.

[30] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, 2006, pp. 153–158.

[31] R. Beverly, A. Berger, and G. Xie, "Primitives for active internet topology mapping: Toward high-frequency characterization," in *Proceedings of the 10th annual conference on Internet measurement*. ACM, 2010, pp. 165–171.

[32] Y. Tian, R. Dey, Y. Liu, and K. Ross, "China's internet: Topology mapping and geolocating," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2531–2535.

[33] K. Chen, D. Choffnes, R. Potharaju, Y. Chen, F. Bustamante, D. Pei, and Y. Zhao, "Where the sidewalk ends," *Proc. ACM CoNEXT*, 2009.

[34] P. Mérindol, B. Donnet, J. Pansiot, M. Luckie, and Y. Hyun, "Merlin: Measure the router level of the internet," in *Next Generation Internet (NGI), 2011 7th EURO-NGI Conference on*. IEEE, 2011, pp. 1–8.

[35] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J. Pansiot, "Topology discovery at the router level: a new hybrid tool targeting isp networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1776–1787, 2011.

[36] University of Oregon Route Views Project. [Online]. Available: http://www.routeviews.org

[37] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (in) completeness of the observed internet as-level structure," *IEEE/ACM Transactions on Networking (ToN)*, vol. 18, no. 1, pp. 109–122, 2010.

[38] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, "10 lessons from 10 years of measuring and modeling the internet's autonomous systems," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1810–1821, 2011.

[39] D. Alderson, L. Li, W. Willinger, and J. Doyle, "Understanding internet topology: principles, models, and validation," *Networking, IEEE/ACM Transactions on*, vol. 13, no. 6, pp. 1205–1218, 2005.

[40] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, and R. Mortier, "Network topologies: inference,

modeling, and generation," *Communications Surveys & Tutorials, IEEE*, vol. 10, no. 2, pp. 48–69, 2008.

[41] W. Willinger, D. Alderson, and J. Doyle, *Mathematics and the internet: A source of enormous confusion and great potential*. Defense Technical Information Center, 2009.

[42] L. Li, D. Alderson, J. Doyle, and W. Willinger, "Towards a theory of scale-free graphs: Definition, properties, and implications," *Internet Mathematics*, vol. 2, no. 4, pp. 431–523, 2005.

[43] Y. Shavitt and N. Zilberman, "A geolocation databases study," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 10, pp. 2044–2056, 2011.

[44] I. Poese, S. Uhlig, M. Kaafar, B. Donnet, and B. Gueye, "Ip geolocation databases: unreliable?" *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 53–56, 2011.

[45] V. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4. ACM, 2001, pp. 173–185.

[46] V. Adhikari, S. Jain, Y. Chen, and Z. Zhang, "Vivisecting youtube: An active measurement study," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2521–2525.

[47] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1620–1628.

[48] V. Adhikari, Y. Guo, F. Hao, V. Hilt, and Z. Zhang, "A tale of three cdns: An active measurement study of hulu and its cdns," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012, pp. 7–12.

[49] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, "Web content cartography," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 585–600.

[50] M. Tozal and K. Sarac, "Subnet level network topology mapping," in *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*. IEEE, 2011, pp. 1–8.

[51] B. Augustin, B. Krishnamurthy, and W. Willinger, "Ixps: mapped?" in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009, pp. 336–349.

[52] A. Rasti, N. Magharei, R. Rejaie, and W. Willinger, "Eyeball ases: From geography to connectivity," in *Proceedings of the 10th annual conference on Internet measurement*. ACM, 2010, pp. 192–198.

[53] Firebug. [Online]. Available: https://addons.mozilla.org/en-us/firefox/addon/firebug/

[54] List of national capitals in alphabetical order. [Online]. Available: https://en.wikipedia.org/wiki/List_of_national_capitals_in_alphabetical_order

[55] Public dns servers. [Online]. Available: https://sites.google.com/site/kiwi78/public-dns-servers

[56] IP to ASN Mapping. [Online]. Available: https://www.team-cymru.org/Services/ip-to-asn.html

[57] RIPE Routing Information Service (RIS). [Online]. Available: https://www.ripe.net/data-tools/stats/ris/routing-information-service

[58] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts," *Networking, IEEE/ACM Transactions on*, vol. 14, no. 6, pp. 1219–1232, 2006.

[59] Trilateration Utility for Location IP hosts. [Online]. Available: https://www.slac.stanford.edu/comp/net/wan-mon/tulip/

APPENDIX

Table 2: PoP locations and the Peering ASes with Google

PoP location (City)	**Peering ASes(AS#)**
Zurich	equinix(AS8235)
Washington	level3(AS3356), Cogent(AS174)
Toronto	ORION(AS26677), NetTorix(AS3303)
Sydney	AARNET(AS7575)
Stockholm	TELIANET(AS1299)
Seattle	WASH-NSF(AS101), SHAW(AS6327)
Sao paolo	ABPTT(AS26162)
San francisco	level3(AS3356)
Paris	GBLX(AS3549)
New York	level3(AS3356), GBLX(AS3549), MFNX MFN(AS6461)
Miami	TERREN-2(AS23148), level3(AS3356), FLORIDANET(AS11096)
Los Angeles	CSUNET-NW(AS2152), level3(AS3356)
London	TELIANET(AS1299)
Lisbon	RCCN(AS1930)
Hong Kong	HKBN(AS9269)
Frankfurt	TELIANET(AS1299), DFN(AS680)
Dallas	level3(AS3356)
Chicago	NWU(AS103), Cogent(AS174)
Buenos Aires	AR-TLCI-LACNIC(11664)

Table 3: PoP locations and the corresponding Planet Lab sites

PoP location (City)	**Corresponding Planet Lab sites**
Seattle	'University of Washington', 'University of Victoria', 'University of Calgary - Computer Science'
New York	'University of Pennsylvania', 'Stevens Institute of Technology', 'Polytechnic Institute of NYU', 'Rutgers University', 'Orbit', 'Boston University', 'Northeastern University CCIS', 'Yale University'
Dallas	'Rice University', 'Texas AM University', 'University of Colorado at Boulder'
Sao paolo	'Universidade Federal de Minas Gerais', 'RNP - Rio Grande do Sul', 'LARC - University of Sao Paulo', 'C3SL - Centro de Computacao Cientifica e Software Livre ', 'RNP - Para'
San francisco	'HP Labs', 'University of Science and Technology of China'
Miami	'University of Florida - ACIS Lab', 'University of South Florida (CSE)', 'University of Puerto Rico at Mayaguez', 'PlanetLab Colo - CLARA Santiago', 'PlanetLab Colo - CLARA Sao Paulo', 'University of Central Florida - EECS', 'ESPOL University'
Toronto	'University of Toronto', 'McGill University', 'University of Waterloo', 'University of Ottawa', 'Carleton University'
Frankfurt	'Max Planck Institute for Software Systems', 'Petru Maior University of Targu Mures', 'University of Wuerzburg', 'Friedrich-Alexander University Erlangen-Nuremberg', 'Technische Universitaet Muenchen', 'University of Duisburg-Essen'
Los Angeles	'University of California at San Diego', 'University of California at Los Angeles', 'Cal Poly State University SLO', 'University of Science and Technology of China', 'Palo Alto Research Center'
Zurich	'ETH Zuerich - Computer Science', 'ETH Zuerich', 'University of Basel, 'University of Neuchatel', 'University of Zurich
Chicago	'Northwestern University (Illinois)', 'Washington University in St Louis'
Washington	'Case Western Reserve University', 'University of Massachusetts Dartmouth'
Buenos	'Universidad de Buenos Aires'
London	'University College London', 'Petru Maior University of Targu Mures', 'University of Cambridge', 'Imperial College London', 'Aston University', 'University of Essex', 'University of St Andrews'
Sydney	'Monash University'
Stockholm	'Petru Maior University of Targu Mures', 'Telefonica Research
Lisbon	'CCTC / Universidade do Minho', 'IT/ISCTE-IUL'
Hong Kong	'The Hong Kong Polytechnic University'
Paris	'Alcatel-Lucent Villarceaux'

Table 4: Google servers location

Google server DNS name pattern	Location From Airport code	Lat from triangulation	Lon from triangulation	AiportCode Available	Triangulation available
atl##s##-in-f#.1e100.net	Atlanta, GA	33.7952	-84.3248	X	X
bud##s##-in-f#.1e100.net	Budapest, Hungary	51.5722	-1.3099	X	X
eze##s##-in-f#.1e100.net	Buenos Aires, Argentina			X	
ord##s##-in-f#.1e100.net	Chicago, IL			X	
dfw##s##-in-f#.1e100.net	Dallas/Fort Worth, TX	29.834	-95.4342	X	X
del##s##-in-f#.1e100.net	Delhi, India			X	
den##s##-in-f#.1e100.net	Denver, CO	41.8776	-87.6272	X	X
fra##s##-in-f##.1e100.net	Frankfurt, Germany			X	
lhr##s##-in-f#.1e100.net	London, United Kingdom - Heathrow	51.5722	-1.3099	X	X
lax##s##-in-f#.1e100.net	Los Angeles, CA	34.0522	-118.244	X	X
mad##s##-in-f#.1e100.net	Madrid, Spain	39.5	-8	X	X
mrs##s##-in-f#.1e100.net	Marseille, France	45.6486	13.78	X	X
mia##s##-in-f#.1e100.net	Miami, FL - International			X	
mil##s##-in-f#.1e100.net	Milan Italy			X	
nuq##s##-in-f#.1e100.net	Moffett Field Airport	37.4178	-122.172	X	X
bom##s##-in-f##.1e100.net	Mumbai, India	6.9354	79.847	X	X
muc##s##-in-f#.1e100.net	Munich, Germany	51	9	X	X
lga##s##-in-f#.1e100.net	New York, NY - La Guardia	40.6944	-73.9906	X	X
kix##s##-in-f#.1e100.net	Osaka, Japan - Kansai Intl	34.75	135.533	X	X
par##s##-in-f#.1e100.net	Paris, France - All airports			X	
gru##s##-in-f#.1e100.net	Sao Paulo, Brazil - Guarulhos Intl			X	
sea##s##-in-f#.1e100.net	seattle			X	
sin##s##-in-f#.1e100.net	Singapore, Singapore	1.3667	103.8	X	X
arn##s##-in-f#.1e100.net	Stockholm-Arlanda			X	
syd##s##-in-f#.1e100.net	Sydney, New South Wales, Australia			X	
nrt##s##-in-f#.1e100.net	Tokyo, Japan - Narita	35.6895	139.692	X	X
yyz##s##-in-f#.1e100.net	Toronto, ON			X	
iad##s##-in-f#.1e100.net	Washington DC - Dulles			X	
oa-in-f###.1e100.net		29.834	-95.4342		X
ee-in-f###.1e100.net		53.3431	-2.6407		X
wi-in-f###.1e100.net		51.5722	-1.3099		X
qa-in-f##.1e100.net		33.7952	-84.3248		X
bk-in-f##.1e100.net		51	9		X
fa-in-f###.1e100.net		51	9		X
lb-in-f###.1e100.net		51	9		X
vb-in-f###.1e100.net		33.7952	-84.3248		X
qe-in-f##.1e100.net		33.7952	-84.3248		X
ve-in-f##.1e100.net		33.7952	-84.3248		X
ob-in-f###.1e100.net		29.834	-95.4342		X
wg-in-f###.1e100.net		51.5722	-1.3099		X
pb-in-f###.1e100.net		47.6103	-122.334		X
hg-in-f###.1e100.net		22.2833	114.15		X
gh-in-f###.1e100.net		28.5663	-81.2608		X
vc-in-f###.1e100.net		33.7952	-84.3248		X
wb-in-f###.1e100.net		51.5722	-1.3099		X
gg-in-f##.1e100.net		33.7952	-84.3248		X
yh-in-f###.1e100.net		28.5663	-81.2608		X
yn-in-f###.1e100.net		33.7952	-84.3248		X
ie-in-f###.1e100.net		42.2923	-83.7145		X
la-in-f##.1e100.net		51	9		X
tb-in-f###.1e100.net		25.0392	121.525		X
iy-in-f###.1e100.net					
ww-in-f###.1e100.net					
tx-in-f##.1e100.net					
pz-in-f##.1e100.net					
yx-in-f##.1e100.net					
hx-in-f##.1e100.net					
ey-in-f###.1e100.net					