YouTube Traffic Dynamics and Its Interplay with a Tier-1 ISP: An ISP Perspective

Vijay Kumar Adhikari, Sourabh Jain, and Zhi-Li Zhang
Department of Computer Science & Engineering, University of Minnesota
Minneapolis, MN
viadhi@cs.umn.edu, sourj@cs.umn.edu, zhzhang@cs.umn.edu

Abstract

In this paper we conduct an extensive and in-depth study of traffic exchanged between YouTube data centers and its users, as seen *from the perspective of a tier-1 ISP* in Spring 2008 after YouTube was acquired by Google but before Google did any major restructuring of YouTube. Using flow-level data collected at multiple PoPs of the ISP, we first infer where the YouTube data centers are located and where they are connected to the ISP. We then deduce the load balancing strategy used by YouTube to service user requests, and investigate how load balancing strategies and routing policies affect the traffic dynamics across YouTube and the tier-1 ISP.

The major contributions of the paper are four-fold: (1) we discover the surprising fact that YouTube does not consider the geographic locations of its users at all while serving video content. Instead, it employs a location-agnostic, *proportional* load balancing strategy among its data centers to service user requests from all geographies; (2) we perform in-depth analysis of the PoP-level YouTube traffic matrix as seen by the ISP, and investigate how it is shaped by the YouTube load balancing strategy and routing policies utilized by both YouTube and the ISP; (3) with such knowledge, we develop a novel method to estimate *unseen* traffic (i.e. traffic that is carried outside the ISP network) so as to "complete" the traffic matrix between YouTube data centers and users from the customer ASes of the ISP; and 4) we explore "what if" scenarios by assessing the pros and cons of alternative load balancing and routing policies. Our study sheds light on the interesting and important interplay between large content providers and ISPs in today's Internet.

Categories and Subject Descriptors

C.2.4 [**Distributed systems**]: Distributed applications; C.4 [**Performance of systems**]: Performance attributes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IMC'10, November 1–3, 2010, Melbourne, Australia.
Copyright 2010 ACM 978-1-4503-0057-5/10/11 ...$10.00.

General Terms

Measurement, Performance

Keywords

YouTube, proportional load balancing, unseen traffic estimation

1. INTRODUCTION

A significant portion of today's digital multimedia content is serviced by large content providers such as YouTube (now a subsidiary of Google), Yahoo, Google, and the like. These large content providers often employ several huge data centers – each of which is comprised of thousands of servers – to meet and serve growing user demands. For a variety of reasons, these data centers are typically located in different geographical sites, and connected to one or multiple ISPs at the nearby major Internet "interconnection regions" or Points-of-Presence (PoPs)[1]. In other words, the content serviced by a large content provider typically flows from one of its data centers through one of these PoPs to enter these ISPs, and is then carried to eventually reach various users. The dynamic *inter-dependence* and *interplay* between content providers and ISPs raise many interesting and important questions that have not been adequately studied.

Take YouTube as an example. As the most popular video sharing site on the Internet, YouTube attracts millions of users every day. Given the significant amount of traffic generated by YouTube videos, how YouTube manages its traffic dynamics and performs load-balancing among its data centers will have a considerable impact on the traffic flows across and within ISPs. Further, the (BGP) routing policies employed by both YouTube and ISPs also shape and drive the traffic dynamics between YouTube and those ISPs. Such dynamic *inter-dependence* and *interplay* therefore have significant implications in traffic management to both YouTube and the ISPs. For instance, can an ISP effectively estimate YouTube-specific traffic matrix so as to better manage its traffic dynamics?

In this paper we take a *measurement-oriented* approach to study the YouTube traffic dynamics *from the perspective of*

[1]For example, according to [11], there are eight major "interconnection regions" within the United States – namely, New York, Washington D.C. and Atlanta on the east coast, Dallas and Chicago in the central US, and Los Angles, the Bay Area and Seattle on the west coast – at which content providers typically buy transit from or peer with various ISPs.

a tier-1 ISP, with emphasis on the *interplay* between the two players, its effect on the traffic dynamics across them, and implications in traffic management for both players. Our study is based on sampled flow-level data collected at various PoPs of a tier-1 ISP.

From BGP routing tables, YouTube (AS36561) advertises 3 prefixes, namely, 64.15.112.0/20, 208.65.152.0/22, and 208.117.224.0/19. Using this information, we first extract all flows related to YouTube from the ISP flow-level measurement data. Through reverse DNS look-ups and other analysis, we infer and deduce that YouTube employs *seven* data centers[2] to service user demands, and that it is connected to the tier-1 ISP at six out of the eight Internet "interconnection regions" (or PoPs) mentioned in [11].

Using the extracted YouTube traffic, we perform an extensive and in-depth analysis of the traffic dynamics between YouTube and the ISP, and explore how load balancing strategies and routing policies employed by both players drive and affect the traffic dynamics between them. In the following we provide a brief overview of the major observations and contributions of our study along four inter-related lines of investigations.

Location-Agnostic Load Balancing: We analyze and infer the load balancing strategies used by YouTube to service user requests. We find that YouTube employs a *proportional* load balancing strategy among the seven data centers (as opposed to, say, a locality- or proximity-aware strategy), where the proportionality seems to be determined by the "size" of the data centers, e.g., as measured by the number of public IP addresses seen in the data that are associated with (front-end) video servers at each data center. This proportionality stays fairly constant over time and across different geographical locations (PoPs) of the tier-1 ISP, and holds for both the client-to-YouTube and YouTube-to-client traffic. *This finding is of particular interest, as it provides an important contrast to the findings of several earlier studies on CDNs [7, 9], which show the prevalence of* proximity-based *content distribution among several CDN services.*

Prevalence of Early-Exit Routing in the ISP Network: With knowledge of the YouTube load balancing strategy, we examine how the YouTube traffic flows between various PoPs of the ISP and the six PoP locations where YouTube is connected to the ISP. In other words, we estimate and analyze the *PoP-level YouTube traffic matrix* as seen by the ISP. Due to the routing asymmetry, we consider the client-to-YouTube and YouTube-to-client traffic separately. We find that the PoP-level *client-to-YouTube* traffic matrix is highly *geographically biased* in the sense that the client-to-YouTube traffic originated from a source PoP always leaves the ISP and enters YouTube at the "closest" of the six destination PoPs, *regardless of* which YouTube data center the traffic is destined to. For instance, the client-to-YouTube traffic originated at a source PoP near New York City, despite being proportionally load balanced across seven YouTube data centers, always exits the ISP and enters YouTube at the New York PoP itself. This observation suggests that YouTube somehow has to carry the client traffic from New York across its "internal" network to the destination data center. On the other hand, the *YouTube-to-client* traffic originated from six YouTube data centers always enters the ISP at the corresponding PoP that they are closest to. For instance, the YouTube-to-client traffic originated from the YouTube New York data center always enters the ISP at the New York PoP. In particular, the YouTube-to-client traffic from the Miami data center is *not* carried by the ISP, i.e., "unseen" by the ISP. This suggests that the YouTube-to-client traffic from the Miami data center is carried by another ISP to the clients. The asymmetry in the client-to-YouTube and YouTube-to-client traffic matrices can be attributed to the BGP routing policies such as "early exit" and route preferences set by the ISP and YouTube.

Estimating Unseen Traffic: Despite the highly asymmetric nature of the client-to-YouTube and YouTube-to-client traffic, the proportionality of traffic split across YouTube data centers (when seen by the ISP) still holds, even when we zero in on a specific customer AS (or prefix) of the ISP. This observation leads us to develop a novel method to estimate the *unseen* YouTube traffic (traffic that is being carried outside of *ISP-X* network), and "complete" the YouTube traffic matrices between YouTube data centers and client ASes of the ISP. This ability to estimate unseen traffic is useful and important both in theory and in practice: It allows us to infer and estimate the total traffic demand matrices between YouTube and various customer ASes of the ISP; it also enables an ISP to estimate the "new" traffic demand and prepare for the potential impact on its network, when routing changes lead YouTube to re-route this unseen traffic through the ISP instead.

Better Understanding of Interplay through "What If" Analysis: Using the YouTube traffic matrices estimated above, we investigate several "what if" scenarios to examine the pros and cons of different load balancing strategies and routing policies, and the impact of the resulting traffic dynamics on both YouTube and the ISP. For example, we find that due to uneven traffic demand distribution across the geographical locations, a locality-aware data center selection strategy [9], while may reduce the overall video download latency for users, can also lead to un-balanced loads across different data centers, especially when the capacities of data centers do not match the geographical load distribution. This plausibly explains why YouTube prefers a location-agnostic, proportional load balancing strategy, as server-load and network bandwidth are likely more critical than latency to the performance of video download/streaming services. We also explore how selective route announcements can shift traffic between YouTube and the tier-1 ISP, and how the ISP can prepare for such changes in YouTube traffic dynamics using the unseen traffic estimation.

To our best knowledge, our paper provides the first extensive study of traffic exchanged between YouTube data centers and users, as seen *from the perspective of a large (tier-1) ISP*. More importantly, while YouTube is one of many (albeit one of great importance) large content providers on the Internet, our work provides a valuable case study that demonstrates the importance of understanding the interplay between large content providers and ISPs. In particular, our study illustrates that the load balancing strategies employed by a content provider (be it proportional load-

[2]As inferred based on DNS names as well as traceroutes, the seven YouTube data centers are located respectively in New York City, Ashburn (near Washington D.C.), Miami, Chicago, Dallas, Los Angles, and the San Jose Bay Area. YouTube is connected to the tier-1 ISP in question at six PoP locations: New York City, Washington D.C., Chicago, Dallas, Los Angles, and the San Jose Bay Area.

balancing, locality-aware, latency-aware, or other more sophisticated strategies) *and* (BGP) routing policies can have a significant impact on the traffic dynamics across content providers and ISPs, and have important implications in traffic management and performance engineering for both content providers and ISPs. With the increasing prominence of data centers in content delivery and emergence of cloud computing, we believe that the interplay between large content (or application service) providers and ISPs will have broad ramifications in a number of key areas such as traffic engineering, network management, business agreements and economic models. Our work is only a first attempt in addressing this broad set of questions.

Caveat. The study presented in this paper provides a snapshot of YouTube traffic as seen by the tier-1 ISP in Spring 2008, shortly after Google acquired YouTube. Since then, Google has been expanding and restructuring YouTube. In Appendix A, we discuss some of the things that have changed after we collected the data used in this study and argue that our methodology and the insights gained from this work still apply to today's YouTube traffic.

The remainder of the paper is structured as follows. In Sec. 2 we overview the YouTube video delivery framework and datasets used in our study. In Sec. 3 we present the method used for inferring YouTube data centers, and in Sec. 4, we analyze and deduce the load balancing strategies used by YouTube among its data centers. In Sec. 5 we study the YouTube traffic matrices as seen by the ISP, and investigate the effects of load balancing and routing policies. In Sec. 6 we present a novel method for estimating the *unseen* traffic, and in Sec. 7 we explore "what-if" scenarios. The paper is concluded in Sec. 8.

2. OVERVIEW & RELATED WORK

2.1 A Quick Overview of YouTube

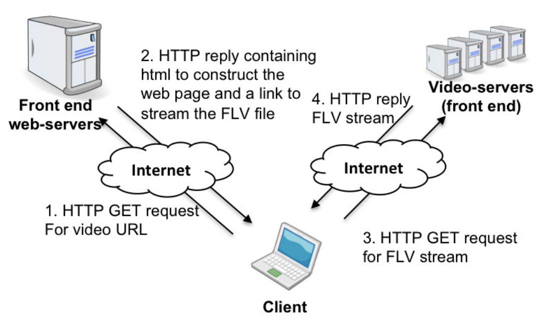


Figure 1: Overview of YouTube video delivery framework.

YouTube is the most popular video serving website that serves user-generated and other videos. It lets any user upload videos in a number of formats. YouTube then converts them in flash-video format and makes them available for viewing. Users with a browser that have Adobe Flash player (or some other compatible player) can watch those videos on YouTube.com or some other sites that embed those videos.

Fig. 1 schematically depicts a typical sequence of steps involved when a user watches a YouTube video. When a user goes to `www.youtube.com` to watch a video, a HTTP request is sent to one of the servers corresponding to `www.youtube.com`, which we refer to as the `(front-end) web-servers`, and a web page is returned that also contains a place-holder for the video. When the user clicks on the video or when the video starts playing automatically, the front-end web server instructs the browser (or the flash plug-in inside the browser) to stream video from another server within one of the YouTube data centers, which actually serves the Flash video content. We refer to these servers as the `(front-end) video-servers` – they are publicly visible in the sense each is assigned a public IP address (and a DNS name). Hence, to watch any YouTube video, a user's browser typically has to communicate with both one of the three front-end web servers and one of many (front-end) Flash video-servers. All parts of this communication - including the video streaming - happens over HTTP.

2.2 Datasets and General Statistics

In our study we use sampled Netflow records collected by a tier-1 ISP, which we will refer to as *ISP-X*, at various PoPs in the US and Europe. The collected flow data is augmented with BGP routing updates and other relevant information such as *AS Paths*, source and destination *IP prefixes* advertised by the neighbor ASes, *egress router* from which the destination prefix is learned (hence this is the router at which the flow exits *ISP-X* network), and so forth. The augmented routing information allows us to attribute the source and destination IP addresses contained in the IP headers to the source/destination ASes and the (specific) prefixes they own and announce via BGP. For our study, we use two flow datasets, each collected at 44 PoPs in US and Europe of the ISP for three consecutive days in two separate weeks in Spring, 2008. Using these datasets, we then extract all YouTube-related traffic, i.e., all flows containing IP addresses (either as source or destination IP) belonging to the 3 prefixes announced by YouTube.

Table 1 summarizes the basic statistics for the first dataset, where the statistics for the *client-to-YouTube* traffic is listed in the second column, and those for *YouTube-to-client* traffic in the last column. (The statistics for the second dataset is similar; we omit them here for lack of space.) In total, there are approximately 172 million sampled flows containing YouTube IP addresses, out of which 54 million flows are from clients (users) to YouTube, and 118 million flows from YouTube to clients. There are no flows from YouTube to YouTube traffic. The difference in the numbers of flows from clients to YouTube vs. YouTube to clients is partly due the *traffic asymmetry* in the two directions: the amount of traffic sent from YouTube to clients are in general far larger than the amount of traffic sent from clients to YouTube. (This traffic asymmetry also manifests in the disparate byte counts of flows in each direction, not shown here, see Sec. 5.) Another contributing factor is the effect of *routing asymmetry*: we see that the number of BGP prefixes for client IPs and the number of client AS numbers in the YouTube-to-client traffic are more than double of those in the client-to-YouTube traffic. Despite the routing and traffic asymmetries, there are ∼3,000 ASes which are seen in both client-to-YouTube and YouTube-to-client traffic, and there are ∼10,000 BGP advertised prefixes that are seen in both directions. Hence these client ASes (or prefixes) use *ISP-X* for connecting to YouTube in both directions.

In addition to the aforementioned augmented flow datasets, we also conduct reverse DNS look-ups and traceroutes to

Table 1: Summary Statistics for Dataset I.

	to YouTube	from YouTube
Total flows	$\sim 54 \times 10^6$	$\sim 118 \times 10^6$
Client IP addresses seen	$\sim 10 \times 10^6$	$\sim 20 \times 10^6$
BGP advertised client prefixes	$\sim 28,000$	$\sim 61,000$
Number of client ASes	$\sim 4,000$	$\sim 9,000$
YouTube IP addresses seen	$\sim 2,000$	$\sim 2,000$

classify YouTube traffic, infer and discover the locations of YouTube data centers, etc. We also collect several gigabytes of *tcpdump* [8] data by playing hundreds of randomly selected YouTube videos on a couple of our lab machines. Collected data includes IP, TCP, HTTP headers and as well as first 200 bytes of the payload. The *tcpdump* data is used to confirm and validate the YouTube traffic characteristics inferred from the sampled Netflow records.

2.3 Related Work

Existing studies of YouTube traffic characteristics have either focused on user behaviors or relied on data collected at end hosts or edge networks. For example, the authors in [3] examine video access patterns, user behavior, popularity life cycle of videos, etc., and compare these characteristics with the traditional web, whereas in [4] the authors explore the "social networks" of YouTube videos. The studies in [6, 16] utilize packet traces collected at campus networks to study the YouTube traffic characteristics from the perspective of an edge network. In a more recent work [12], Fahmy *et al.* analyze and compare the underlying distribution frameworks of three video sharing services, YouTube, Dailymotion and Metacafe, again using the traces collected at edge networks. The focus of the work is to investigate the variation in service delay experienced by users in different geographical locations, and when accessing videos of different popularity and ages. In contrast, using the flow-level data collected at multiple PoPs of a tier-1 ISP, we study the interplay between YouTube and the ISP by investigating the effects of load balancing strategies and routing policies (of both YouTube and the ISP) on the YouTube traffic dynamics. Our work also differs from existing studies (see, e.g., [1, 5, 14]) on characterizing large-scale *Content Distribution Networks* such as Akamai. With increasing reliance on large data centers for content delivery, our work sheds light on the interesting and important interplay between large content providers and ISPs.

3. CLASSIFICATION & LOCALIZATION OF YOUTUBE SERVERS

As explained in Sec. 2.1, YouTube has a two-tier video delivery framework, where servers can be divided into mainly two categories, *front-end* and *video-servers*. Although it might not be very difficult to get a list of data centers YouTube uses, it is a non-trivial task to identify the geographic locations and the roles played by individual YouTube servers in the video delivery framework. In this section we use the Netflow data along with DNS records to classify YouTube server IP addresses into these categories. We also use another dataset collected using the *tcpdump*, by playing a large number of videos on couple of client machines, to validate the classification. We explain how we map YouTube server IP addresses to different locations using reverse DNS mapping and traceroutes from multiple vantage points.

3.1 Classifying YouTube Server IP Addresses

We see a total of 2093 YouTube IP addresses in the datasets, and perform reverse DNS look-ups on each of them. Using the host names thus resolved, we classify YouTube IP addresses and corresponding flows into three categories: i) 3 YouTube IP addresses (i.e., *front-end web servers*) whose host names are all mapped to `www.youtube.com`; the corresponding flows are referred as the *web front-end* flows. ii) more than 2000 YouTube IP addresses (83% of the total YouTube IP addresses) whose host names are of the format `loc-`*v*`xx.loc.youtube.com` (e.g., `dal-`*v*`26.dal.youtube.com`), where `loc` is one of the 8 city codes, `ash, chi, dal, lax, mia, nyc, sjc, sjl`, and `xx` is an integer. We deduce and later validate that these host names correspond to *front-end (Flash) video servers* within various YouTube data centers. iii) The remaining IP addresses are either mapped to host names with other formats (roughly 5%), containing keywords such as *smtp*, *dns, db, webcam*, or are not resolvable, i.e., with no public host names (roughly 11%). For flows associated with these IP addresses, many of them are either UDP, or are associated with TCP port numbers other than 80. Hence we deduce that these flows are unrelated to YouTube video delivery, and hence ignore them in the remainder of our study. For flows belonging to the first two categories, we further classify them based on the directions of the flows: *client-to-YouTube*, or *YouTube-to-client*.

Validating YouTube Server IP Address Classification using *tcpdump* Data. To validate the YouTube IPs and flow classification presented above, we also collect *tcpdump* data (packet traces) at end hosts by playing hundreds of randomly selected YouTube videos on these hosts. By examining the payload, we obtain the *ground-truth* regarding the role of the YouTube IP addresses seen in the *tcpdump* data. As above, we classify the flows into four (sub-)categories and characterize the overall traffic characteristics of each category: *i) Clients to (web) front-ends:* These packets carry the HTTP requests to front-ends for downloading various objects in the web-page. These packets are likely to be smaller but of variable sizes. *ii) Clients to Video servers:* These packets mainly carry the acknowledgments for the data sent by the video servers. These packets generally contain a smaller number of bytes. *iii) Front-ends to clients:* These flows carry packets mostly with HTML and javascripts etc which describes the downloaded web-page. These packets are likely to be larger packets, with varying number of bytes because of the varying sizes of the web objects. *iv) Video servers to clients:* These flows carry the video content, therefore are likely to be containing a constant but large number of bytes.

For flows of each category, we obtain the overall traffic characteristics of the flows obtained using the *tcpdump* data collected at the end hosts, and compare them with those obtained flows within the same category in the ISP (sampled) flow datasets. As an example, in Fig. 2, we plot the cumulative distribution of the packet sizes of flows in each category computed using the *tcpdump* data as well as the same distribution computed using the ISP datasets. The distribution of the former is shown as the dashed line, and the latter the solid line. We see that the two distributions in

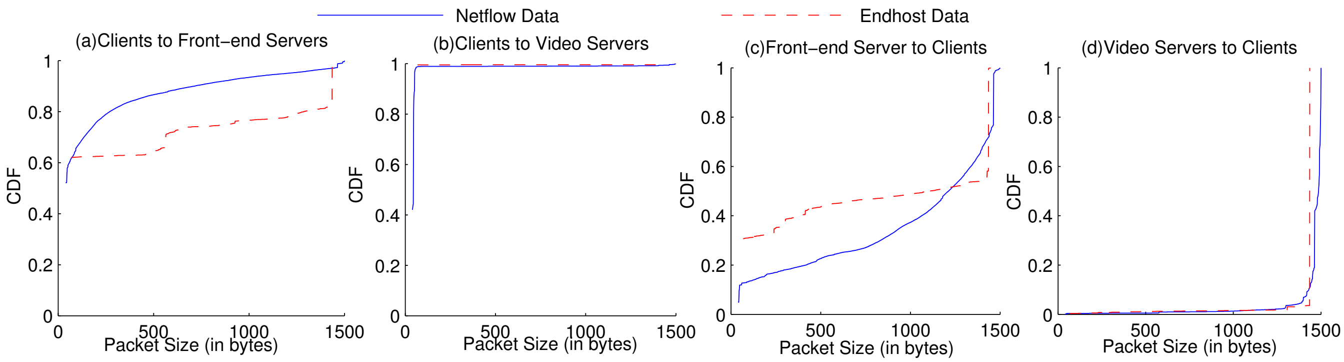


Figure 2: Packet size distribution.

each category match quite well, and are visibly distinct from those in the other categories. These results corroborate and confirm that the YouTube IP address and flow classification presented is indeed valid.

3.2 Locating YouTube Servers

The host names of video server IP addresses discovered through reverse DNS look-ups seem to suggest that there are eight data centers. We perform traceroutes from multiple vantage points to geo-locate and verify the locations for the IP addresses. We find that traceroute queries corresponding to the IP addresses located in the same city produce the same last-hop IP addresses. This traceroute-based analysis also reveals that the two city codes `sjc` and `sjl` point to the same (data center) location, namely within the San Jose Bay Area. We therefore group these two together as a single data center location. This yields a total of seven data centers, as shown in Table 2. The last column of the table also provides the percentage of video server IP addresses belonging to each data center, as seen in the ISP datasets.

We further associate the data center locations (city codes) with the PoP locations of the ISP. More specifically, we compare these origin PoP locations for each YouTube IP address with our 3-letter city code based classification. Except for those YouTube video server IP addresses containing the city code `mia`, in all other cases flows containing (as the source IP address) YouTube video server IP addresses originate from either a source PoP with the same city code or from a source PoP that is nearby. For instance, `nyc-v24.nyc.youtube.com` appears as a source IP address only in flows from New York PoP of the ISP, whereas `ash-v22.ash.youtube.com` appears as a source IP address only in flows from Washington D.C. PoP of the ISP. Furthermore, flows with the source IP addresses containing the city code `sjc` and `sjl` are only seen at the San Jose Bay Area PoP of the ISP. This analysis also reveals that YouTube is connected with the ISP at six PoP locations, Washington DC, New York City, Chicago, Dallas, Los Angles and San Jose.

Lastly, in terms of the three front-end web server IP addresses, they only show up in flows (as source IP addresses) originating from the San Jose PoP. Moreover, other YouTube IP addresses within the same /24 block are also mapped to the San Jose PoP. Traceroute analysis also reveals that these three front-end web server IP addresses are located within the San Jose Bay Area. Hence we deduce that all three front-end web servers are located within the San Jose Bay Area.

Table 2: Number of video servers at different locations

Code	City	% of IP addresses
ash	Ashburn, VA	24.77
chi	Chicago, IL	17.27
dal	Dallas, TX	07.89
lax	Los Angeles, CA	17.33
mia	Miami, FL	7.37
nyc	New York City	7.26
sjc, sjl	San Jose Bay Area	18.07

4. LOAD BALANCING AMONG YOUTUBE DATA CENTERS

In this section we analyze and infer the load balancing strategy used by YouTube among its data centers to service user requests. The key discovery we make is that YouTube employs a *proportional* load balancing strategy among the seven data centers and not based on the geographical proximity to clients, where the proportionality seems to be determined by the "size" of the data centers; this is in contrast to several studies [7, 9] on CDNs which typically exploit *proximity-based* server selection strategy to reduce end-to-end latency[3].

4.1 Proportional Load Balancing of Client-to-YouTube Traffic

We now examine how client-to-YouTube traffic is distributed among the seven YouTube data centers. Our initial hypothesis is that YouTube does proximity-aware content distribution meaning clients are served videos primarily from the data-center that results in lowest latency. To our surprise, we found that this was not the case. For instance, when we examined a "/24" prefix belonging to New York Public Library, it was exchanging only about 7% of the total flows with YouTube NYC data-center. Similarly, another pre-

[3]In Sec. 7 we will discuss the pros and cons of using *proportional* load-balancing vs. *proximity-aware* load-balancing strategies. We also note that the authors in [9] find that simply relying on *proximity* for serving client requests lead to poor CDN performance for some users; the quality of paths should be also taken into account.

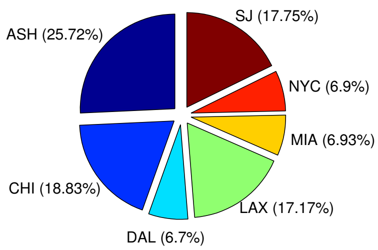


Figure 3: Distribution of client traffic to YouTube data centers.

fix owned by Illinois Century Network was only exchanging 19% of its traffic with YouTube Chicago data-center. It was also very clear that the flows were not being equally divided among data-centers because YouTube Ashburn data-center was receiving the highest share of traffic from both the networks and was significantly higher from traffic shares that other data-centers were getting. When we examined the traffic distribution more closely, we noticed that the traffic was being distributed at a fixed proportion to YouTube data-centers from all the *ISP-X* PoPs irrespective of which PoP was geographically closer to which data-center.

The distribution of the overall client-to-YouTube traffic is shown in Fig. 3. We can see that the client-to-YouTube traffic is clearly not equally divided among the data centers. In particular, we see that the ASH data center receives slightly more than a quarter of the traffic, while the LAX, CHI, and DAL data centers receive about 17% of the total traffic. On the other hand, the DAL, MIA and NYC data centers receive only ∼7% of the total traffic each. This is very similar to what we observed in case of the two prefixes we discussed above.

Fig.3 shows how the total traffic is divided. Next, we look at how traffic coming from individual source PoPs of *ISP-X* is distributed. In Fig. 4, we show how client-to-video server traffic is distributed among 7 YouTube data centers for traffic originating at each *ISP-X* PoP. In this figure x-axis shows different PoP for *ISP-X*, and y-axis shows the fraction of total traffic going to different YouTube data centers. As seen in this figure, the fraction of traffic received by each YouTube data center from each PoP is the same for all PoPs. It implies that although PoPs of *ISP-X* and YouTube data centers are located at geographically diverse locations, they do not have any preferences to each other in terms of network/geographic "proximity". More specifically, traffic coming from clients is distributed to 7 YouTube data centers according to *fixed proportions*. We have also verified that these proportions hold at various other levels such as over time, at AS level, and individual IP prefix level (plots not shown due to space limitations).

4.2 Load Balancing among Three Web Servers

We now analyze how YouTube performs load balancing among the three front-end web servers. Since all the web-servers are located inside San Jose data-center, all the traffic to front-end web servers have to come to San Jose. An interesting but not so surprising discovery is that the traffic is equally distributed among those 3 front-end server IP addresses. This observation holds true at different time granularity, e.g., over different hours or over different days, as well as across different spatial granularity, e.g., at the level of individual PoPs of the ISP or with respect to individual client ASes. These observations are shown in Fig. 5, where Fig. 5(a) plots the traffic distribution among the three front-end web servers over different hours, Fig. 5(b) plots the traffic distributions among the three front-end web servers at various PoPs of the ISP, and Fig. 5(c) plots the traffic distributions among the three front-end web servers at various (large) client ASes of the ISP. That the traffic is equally distributed among the front-end web servers is expected, as YouTube front-ends employs the round-robin DNS resolution [2] for mapping `www.youtube.com` to the IP addresses of front-end web servers.

4.3 Proportional Load Balancing of YouTube-to-Client Traffic

In the discussion so far we used only one direction of the traffic. i.e, traffic going from clients to YouTube data centers. Our findings show that this traffic is proportionally distributed among various YouTube data centers. In the following, we analyze the traffic from the reverse direction, i.e., from YouTube data centers to clients. Comparing distributions in these two directions is inherently challenging because of asymmetric routing in the Internet. Routing in the Internet is not symmetric, that is the network path taken by the traffic going from a client to one of the YouTube data centers may not be the same as the path taken by the reply traffic coming from the YouTube data center to the client. Furthermore, since the network flow data available to us is only collected at various PoPs of the *ISP-X*, it may not have the YouTube to client traffic for some of the client-to-YouTube traffic, as the reply traffic from YouTube can go through different ISPs. Therefore, in order to analyze how the traffic from YouTube data centers to various *ISP-X* PoP is distributed, we only consider the IP prefixes for which we see the traffic coming from all the YouTube data center locations. In addition, some of the YouTube data centers may not use *ISP-X* to send the reply traffic to customers at all. In our data, we do not find any reply traffic from Miami YouTube data center. Hence we ignore traffic from Miami YouTube data center in the following analysis.

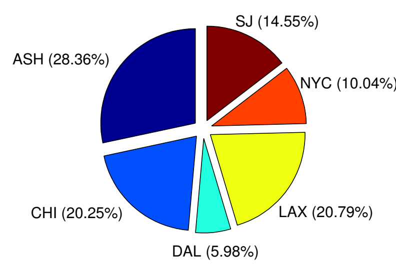


Figure 6: Traffic distribution from YouTube data centers to clients.

Fig. 6 shows how much traffic is sent by each of the YouTube data center to clients. As seen in this figure, the largest fraction of video traffic is sent by the YouTube ASH data center. While CHI and LAX send 20% of the traffic each, and remaining traffic is sent by the SJ, NYC and DAL data centers.

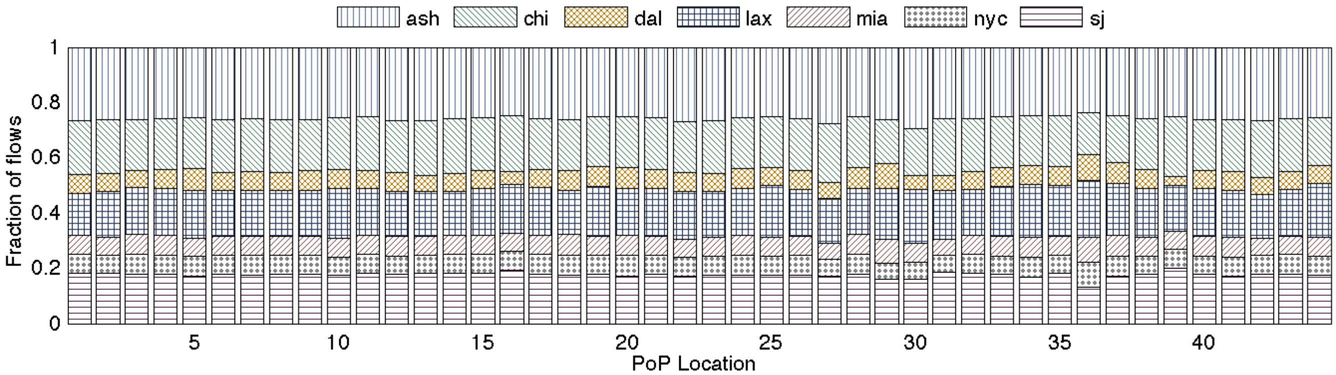


Figure 4: Client traffic distribution to different video serving data centers.

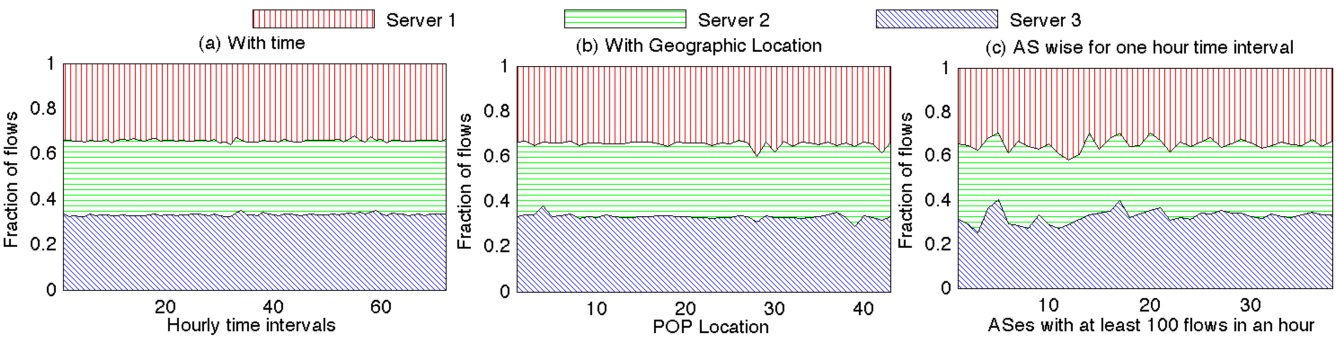


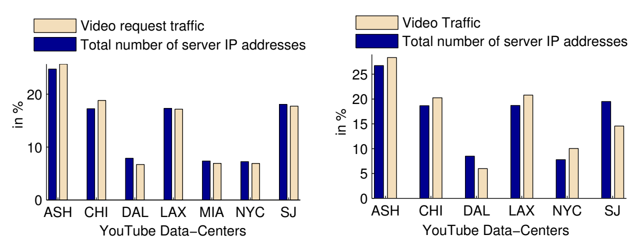
Figure 5: Load balancing among three front-end YouTube servers.

Next, we investigate if the proportional traffic distribution holds for the YouTube data center-to-client traffic at the PoP level. As mentioned before due to asymmetric routing in Internet, for this analysis, we consider traffic for only those IP prefixes which receive traffic from all the YouTube data centers (except Miami data center) through *ISP-X*. Furthermore, we group all the IP prefixes using *ISP-X* PoPs. For this we assign a PoP to each IP prefixes, which is the PoP from where traffic with the corresponding IP prefix enters in the ISP network.

We plot the fraction of traffic sent by each data center to various *ISP-X* PoPs in Fig. 7. In this figure different destination *ISP-X* PoPs are shown on the x-axis, and y-axis shows the fraction of the traffic as number of sampled flows received by the PoP, from a given YouTube data center. As seen in this figure, traffic from different YouTube data center to *ISP-X* PoP is distributed proportionally. In addition, these proportions are the same as the total proportions of the traffic sent by each data center shown in Fig. 6.

4.4 Proportional Load Balancing and "Size" of Data Centers

Our analysis has shown that YouTube does not do a proximity-aware load balancing. Instead, it does load balancing among its data centers in a fixed proportions. As we show next, the distribution of incoming video-requests to its data centers based on the "size" or the "resources" available at those data centers. Since we do not have any access to internal system design of each YouTube data center, we use the number of IP addresses seen at each location as the rough estimate of the size of a data center. We plot the fraction of traffic received by each YouTube data center along with the fraction of IP addresses seen from different data centers in Fig. 8(a). As we see in this figure, the amount of traffic received by each data center is directly proportional to number of IP addresses seen at that location. Furthermore, Fig. 8(b) shows



(a) Client to YouTube traffic (b) YouTube to client traffic

Figure 8: Distribution of traffic between clients and YouTube data centers.

that fraction of traffic sent by different YouTube data centers to *ISP-X* PoPs, is proportional to the fraction of total number of IP addresses seen at the corresponding data center. It shows that *YouTube performs load-sharing among its data centers based on their "size", not location.*

Extracting proportions using SVD: Next, we use the rank-1 approximation using SVD[4] [15] to estimate the pro-

[4]Albeit the proportionality can also be verified directly, the SVD automatically yields the proportionality (or a rank-1 matrix approximation) that best fits the data.

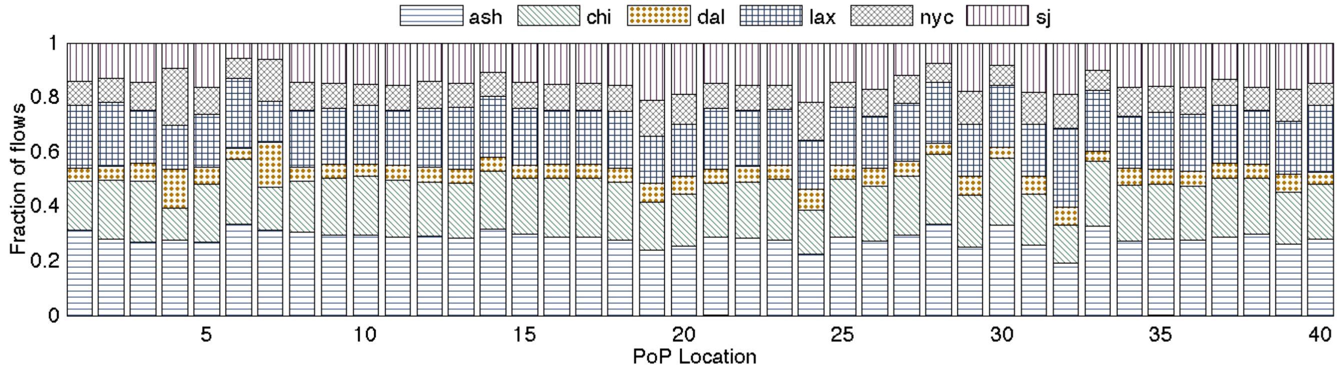


Figure 7: Traffic distribution from video servers to various client PoPs.

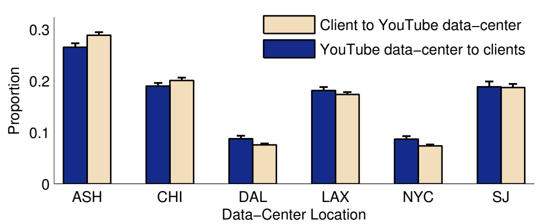


Figure 9: Traffic proportions extracted using SVD.

portions used to distribute the traffic among different data centers. For this analysis, we construct the traffic matrices C and S for client-to-YouTube and YouTubeto-client traffic respectively. Here C_{ij} is the traffic originating from ith /24 IP prefixes which is destined to jth data center, and S_{ij} is the traffic received by ith prefix from jth data center. It is possible that some IP prefixes may not be using *ISP-X* network to reach all the data centers and vice versa, therefore, for the unbiased estimate of proportions, we ignore the /24 IP prefixes which do not send data to all the data centers for the traffic matrix C, and prefixes which do not receive data from all the data centers for the traffic matrix S. Since we do not see traffic coming from YouTube Miami data center, we ignore it in this analysis so that we can compare the proportions seen in both directions of the traffic. We construct a large set of matrices c and s by randomly selecting 1000 rows from matrices C and S respectively to verify if the proportions hold for all the prefixes for different random selection of IP prefixes. We perform the SVD based rank-1 approximation on matrices c and s. Fig. 9 shows the proportions estimated for both client-to-YouTube and YouTube-to-client traffic using matrices c and s. In this figure, black horizontals line on top of each bar show the 95% confidence interval for the mean values shown in the figure. As seen in this figure, traffic is distributed in the same proportions for both matrices, with very little variances.

4.5 Traffic Volume Asymmetry

As we have already observed, the amount of client-to-YouTube flows is much smaller compared to YouTube-to-client flows(see Sec. 3). In this section we further analyze the YouTube-to-client and client-to-YouTube traffic ratios to get a better insights of the YouTube traffic. For this analysis, we define "two-way traffic pairs" as $\langle c2s(p,dc), s2c(p,dc)\rangle$ for different /24 IP prefixes p and YouTube data centers dc. We use term $c2s(p,dc)$ to represent the traffic sent by IP prefix p to dc data center of YouTube. Here we are interested in learning how the ratios $r(p,dc) = \frac{s2c(p,dc)}{c2s(p,dc)}$ are distributed.

Due to asymmetric routing in Internet some of the two-way traffic pairs may contain zero values. Therefore we only consider the pairs with non-zero values in our analysis. Fig. 10(a) shows the distribution of $r(p,dc)$ for all such two-way traffic pairs for the number of bytes exchanged between IP prefix p and data center dc, and Fig. 10(b) shows the distribution of $r(p,dc)$ using number of flows. As seen in this figure, the ratio of the bytes exchanged between IP prefixes and data centers stays between values 40 and 80 for more than 80% of the pairs, on the other hand the ratio of the flows exchanged is most of the times close to 2. *It demonstrates that traffic exchanged between IP prefixes and data centers is fairly consistent, and can be used for estimating the YouTube traffic that is not seen in the ISP-X network.* (See Sec. 6).

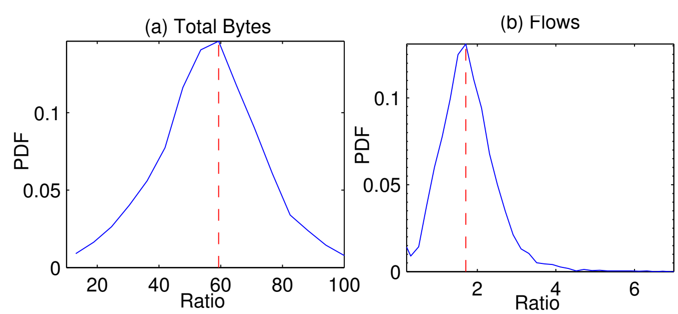


Figure 10: Distribution of ratios for the traffic exchanged between IP prefixes and YouTube data centers.

5. POP-LEVEL YOUTUBE TRAFFIC MATRICES

In Section 4, we made the observation that the YouTube traffic from clients at all locations are divided in a fixed proportion irrespective of whether a data-center is nearer to a client or very far from it. We also saw that the traffic from the data-centers to the clients also follow similar proportional distribution. This behavior of end-to-end traffic matrices naturally raises the question of whether the traffic also gets divided in a similar way inside the *ISP-X* backbone. In this section we look at the *entry PoP to exit PoP traffic-matrix* from the perspective of *ISP-X* with the objective to better understand how the traffic enters *ISP-X* network and gets divided to different exit PoPs.

To study this interaction, we generate *entry-exit* traffic matrices based on the entry and exit PoPs for the YouTube traffic from the perspective of *ISP-X*. We have following two entry-exit matrices based on the direction: (i) from clients to YouTube servers, and (ii) from YouTube servers to clients.

5.1 Single Preferred Exit in Client to YouTube Traffic

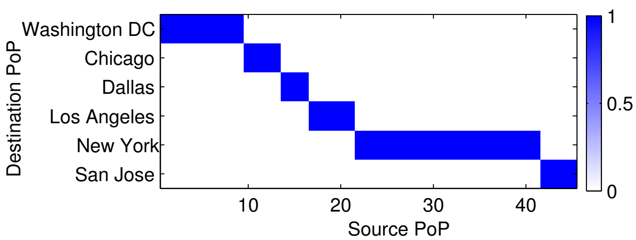


Figure 11: Entry PoPs to exit PoPs traffic distribution from client to YouTube traffic

For the clients to YouTube servers traffic we look at how much traffic enters from an *ISP-X* PoP and which PoP it exits from. We see that the traffic exits from only 7 of the *ISP-X* PoPs where YouTube is present. We also see that for each of the 44 PoPs from which the client to YouTube traffic enters *ISP-X*, all of the traffic from that PoP exits *ISP-X* from exactly one of the 7 exit PoPs. If we represent the entry PoPs as columns of the traffic matrix and exit PoPs as the rows, we see that each column has exactly one non-zero entry. A closer inspection of this matrix reveals that for any PoP P, it has a single exit-pop E_P such that E_P is the geographically closest *ISP-X* PoP where YouTube is present. Clearly, for the seven PoPs where YouTube data centers are present, $P = E_P$.

Next, we compare this entry-exit traffic matrix with the "end-to-end traffic demand and supply" matrices from YouTube's perspective seen in Sec. 4. We see that they are significantly different. For instance, the end-to-end traffic matrix which shows how the total traffic to YouTube video-servers are divided among different data centers, is a rank-one matrix, and follows the gravity model [10]. On the other hand the entry-exit traffic matrix seen by *ISP-X* is a full-rank sparse matrix where each column has exactly one non-zero row. From the point of view of *ISP-X*, traffic flows do not seem to be getting divided to different locations.

To understand the reason behind this difference between the two traffic matrices, we examined BGP routing information. From the BGP data, we see that YouTube aggregates all of its IP addresses in all the data centers in only 3 IP prefixes: 64.15.112.0/20, 208.65.152.0/22, and 208.117.224.0/19. All of these three prefixes are then announced to *ISP-X* at each of the 7 PoPs where YouTube is present. This allows *ISP-X* to do an early exit (or hot-potato) routing [13]. Since all the YouTube prefixes are announced from each of the locations where YouTube is present, *ISP-X* can transfer the YouTube-bound traffic at the exit PoP that is nearest to the source PoP irrespective of the final destination data center for the traffic.

Fig. 11 shows how the traffic entering from different *ISP-X* PoPs exits the ISP network. As we can see PoPs 1 to 9 send all of their YouTube-bound traffic towards Washington DC PoP. Similarly, there are 20 PoPs for which New York is the nearest exit PoP to reach YouTube. It shows that each *ISP-X* PoP has exactly one preferred exit PoP for all the YouTube traffic, irrespective of the final destination data center. We use this mapping of source PoPs to their nearest exit-PoP in Sec. 7 for some of the "what if" analysis.

Traffic carried by YouTube: The early-exit routing employed by *ISP-X* has another implication in terms of how the traffic reaches the final destination data center. Since *ISP-X* hands over all the YouTube-bound traffic to YouTube at local PoPs, YouTube has to deliver this traffic to the final destinations using its own "internal" network.

To estimate the amount of client to YouTube traffic that YouTube has to deliver to its different data centers, we look at how much traffic *ISP-X* hands over to YouTube at each location and how much of it is destined to the local YouTube data center. For instance, from all the clients in nearby PoPs, if *ISP-X* hands over 100 units of traffic to YouTube at New York PoP, then 7%, 7%, 17%, 7%, 18%, 16% and 17% of it are destined to New York, Miami, Los Angeles, Dallas, Chicago, Ashburn and San Jose data centers respectively. We see that only 7 units of traffic is actually delivered locally to New York data center, while remaining 93 units of the traffic still has to be carried somehow to other data centers. Overall, we see that only ∼10% of the total traffic remains local, while remaining ∼90% of the client to YouTube traffic is delivered to the final destination data centers by YouTube using its "internal" network.

5.2 High Locality-bias in YouTube to Client Direction

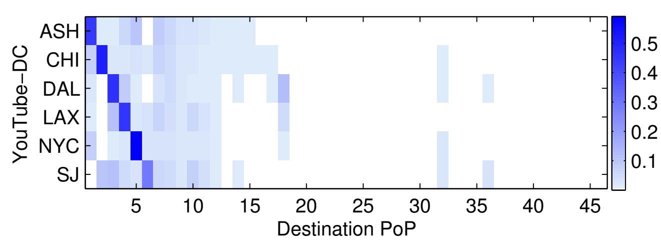


Figure 12: YouTube servers to exit-PoP traffic distribution

Although YouTube uses a proximity-agnostic mechanism to load balance the traffic between YouTube servers and client, the entry-exit traffic matrix at PoP level shows strong locality bias. In Fig. 12, each row represents the distribution of traffic originating from a YouTube data center, which enters the *ISP-X* network through the corresponding local PoP, to all the PoPs as a fraction of the total traffic coming from that data center. Also, in this figure first 6 destination PoP are the ones where the six data centers connect to the ISP in the same order as YouTube data centers shown in y-axis. This figure shows from each PoP from which YouTube to client traffic enters *ISP-X*, about 40% to 60% of the traffic exits *ISP-X* network from the local PoP itself, and only the remaining traffic enters the *ISP-X* backbone.

Not surprisingly, early-exit routing is again the reason behind high locality bias in the YouTube to client traffic. When we look at the BGP information available in the flows, a very large number of prefixes are announced from multiple *ISP-X* PoPs. Therefore, in many cases *ISP-X* selects an exit PoP from a list of PoPs where the destination prefix is announced and in general it prefers the PoP that is geographically closest to the entry PoP.

These results show that even when the content-providers do not do geography-aware load balancing, large ISPs still can still employ early-exit routing and carry less traffic in their backbone.

6. ESTIMATING UNSEEN TRAFFIC

As seen in Sec. 4, the proportionality of traffic split across YouTube data centers (as seen by the ISP) holds at various levels such as over different time intervals or at individual AS or IP prefix level. Furthermore, we showed that ratio of the traffic exchanged between YouTube data centers and individual client prefixes is fairly stable (See Fig. 10). These observations led us to develop a method to estimate the *unseen* and *partially seen* YouTube traffic – namely the (portion of) traffic carried by other ISPs – and "complete" the traffic matrix between YouTube data centers and users from the customer ASes of the ISP.

Basically we have two ways of estimating the unseen traffic: a) based upon the proportional distribution among data-centers, and b) based upon the ratio of client-to-YouTube and YouTube-to-client traffic. These approaches have different error characteristics. The proportions at which the traffic gets distributed to (and from) different data-centers have very small variations compared to the ratio of traffic in two direction (see Figures 9 and 10). Therefore, we try to use the proportionality of traffic distribution to do the estimate as much as possible. Only when its not possible to use that piece of information (such as when we see no client to YouTube data at all), we make use of the ratio of traffic in different directions.

• **Formulating a matrix completion problem:** To facilitate the discussion in this section, we represent the client-to-YouTube traffic matrix by C, where $C_{i,j}$ = Total bytes from ith IP prefix to jth data center, and the YouTube-to-client matrix by S, where $S_{i,j}$ = Total bytes to ith IP prefix from jth data center. Both C and S have unknown entries representing the traffic carried between corresponding client prefix and YouTube data center. To get a complete picture of how much traffic is actually exchanged between an IP prefix and and a YouTube data center including the traffic not carried by *ISP-X*, we need to estimate all such unknown values. The problem therefore can be viewed as a matrix completion problem where some entries are known and some unknown and we use the known entries to compute the values of the unknown entries.

• **Estimation using proportional distribution:** For our matrix completion problem, we first use the proportions at which the YouTube traffic is divided among its 7 data centers. First, we divide the rows of the matrices C and S to form three sub-matrices each: C^f and S^f which consist of all the rows for which all the column entries are known, C^p and S^p which consist of all the rows for which at least one column entry is known, and C^0 and S^0 which consist of all the rows for which none of the column entries are known.

Next we use the proportion $x_1 : x_2 : x_3 : x_4 : x_5 : x_6 : x_7$ (extracted in Sec. 4) to fill the missing entries of C^p as follows. For each row, we first get the known values, say $C^p_{i,j_1}, C^p_{i,j_2}, \ldots, C^p_{i,j_m}$ where $1 \leq m \leq 7$. Each of the unknown entry $C^p_{i,k}$ in that row can be estimated as $\frac{(C^p_{i,j_1}+C^p_{i,j_2}+\ldots+C^p_{i,j_m})\times x_k}{(x_{j_1}+x_{j_2}+\ldots+x_{j_m})}$. We apply the same mechanism to fill the missing entries in S^p by using the proportions.

For filling the values in C^0 and S^0 we use a slightly different approach. Since we do not have any known entries in any of the rows, we exploit the mean ratio between client-to-YouTube and YouTube-to-client traffic to estimate one entry per row for C^0 using the corresponding non-zero value from S^p or S^f. Since all of the IP prefixes in C and S must have either sent or received traffic from YouTube data centers through *ISP-X*, for each completely unknown row in C, the corresponding row in S must have at least one known entry. For each row in C^0, we pick one known field from the corresponding row in S and divide it by the mean ratio to get the corresponding column in C^0 and vice versa for S^0. Finally we combine the rows of C^f, C^p and C^0 to get the completed matrix C' for C, and S' by combining S^f, S^p and S^0 for in-complete initial traffic matrix S.

• **Results:** For the estimate of "unseen" traffic, we first extracted all the IP prefixes from the Netflow data, which were seen either in client-to-YouTube traffic or YouTubeto-client traffic or both. There are around 150,000 such IP prefixes in our Netflow data. Using these prefixes we prepared matrices C and S by extracting the total number of bytes exchanged by these prefixes with different YouTube data centers.

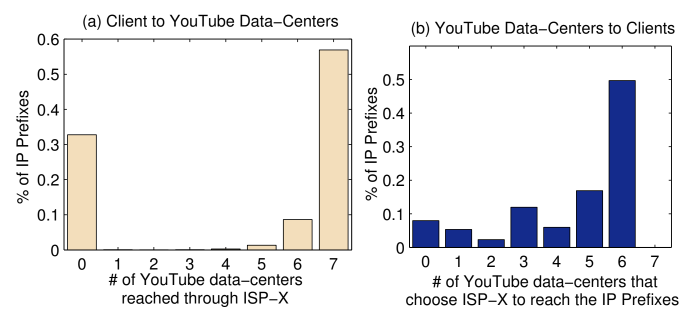


Figure 13: Distribution of IP prefixes on the basis of total number of YouTube data centers reached by them using *ISP-X*

Fig. 13 shows the distribution of IP prefixes seen in the YouTube traffic going through the *ISP-X*'s network, on the basis of the total number of YouTube data centers reached by them using *ISP-X*. In Fig. 13(a) we see that more than 50% prefixes use *ISP-X* to reach all 7 YouTube data centers, on the other hand less than 50% prefixes received traffic from 6 YouTube data centers through *ISP-X*. These figures show that there is a large amount of unseen traffic for these IP prefixes which is delivered through other ISPs.

We show the distribution of estimated unseen traffic for IP prefixes using our methodology in Fig. 14. In this figure, x-axis shows the IP prefixes and y-axis shows the corresponding unseen traffic as a fraction of total traffic received or sent by these prefixes to all the YouTube data centers. Moreover, for the client-to-YouTube data centers traffic we

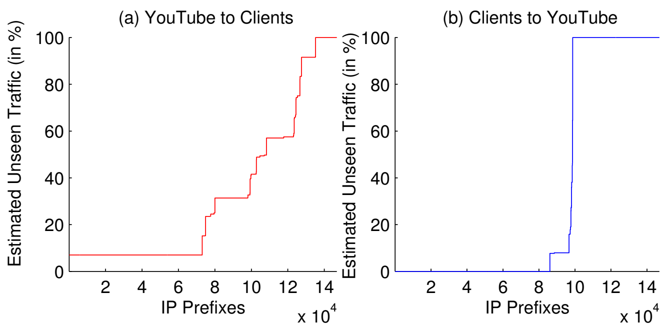


Figure 14: Distribution of estimated unseen traffic for IP prefixes

Table 3: Summary of traffic estimation. (We hide the original unit here due to the anonymity reasons.)

	to YouTube	from YouTube
Total seen traffic	0.59	39.7
Estimated unseen traffic	0.34	20.4

see that 36.5% of the estimated total traffic is not seen by *ISP-X*. Similarly, for the YouTube-to-client traffic 34% of the estimated total traffic is unseen. Overall the percentage of unseen traffic is 34.1% of the estimated total traffic. We summarize these results in Table 3.

We conclude from above that *ISP-X* does not see a huge portion of the traffic exchanged by its customer ASes with YouTube. Therefore, any changes in the routing policy used by its customer ASes and YouTube can have significant impact on the *ISP-X*'s network. Furthermore, this estimation of the unseen traffic can now enable the *ISP-X* to estimate the "new" traffic demand and prepare for the potential impact on its network, when routing changes lead to YouTube to re-route unseen traffic through the ISP instead.

7. "WHAT IF" ANALYSIS

In Section 4, we observed that YouTube does a location-agnostic proportional load-balancing. We also saw in Section 5 that YouTube announces all-inclusive large prefixes from all the locations where it peers with *ISP-X*. Finally, we were able to estimate some of the traffic that is exchanged between clients and YouTube that *ISP-X* does not see. These results give us insights on what is happening now. However, since the dynamics of the network change over time, it is natural to ask what happens if things change. In this section we use the insights gained about YouTube and its traffic characteristics with respect to *ISP-X* to investigate some of the "what if" scenarios to examine the pros and cons of different load-balancing strategies and routing policies, and the impact of the resulting traffic dynamics on both YouTube and the ISP.

• **Scenario 1: What if YouTube employs a location aware load balancing among its data centers?**
We investigate the scenario when instead of being location-agnostic, YouTube employs a "location based" load balancing scheme. One way YouTube can achieve this is by configuring its front-end web servers in such a way that they direct the clients' video requests to a data center that is nearest to the location of the requesting client.

We already know when the client to YouTube traffic enters *ISP-X* network from a PoP it exits from exactly one exit PoP where YouTube is attached. We will use this information to assign each PoP to its nearest YouTube data-center. For instance, YouTube data center near the New York PoP will be the only data center that will deliver video traffic to all the PoPs that use New York PoP as the exit PoP for YouTube bound traffic. We then evaluate how it affects the *ISP-X*'s traffic matrix and also the load at different YouTube data centers.

Effect on the ISP traffic matrix: Since all the client PoPs will fetch videos from the nearby YouTube data center in this scenario, the traffic that is carried in the ISP backbone will be reduced significantly. For instance, when we computed the amount of traffic that remains local at the source PoP under this approach, we see that about 44.5% of overall YouTube to client traffic does not even enter the ISP backbone compared to the same number for location-agnostic load balancing. Moreover, it eliminates most of the "long-distance" communication for the ISP. For example, Los Angeles data center will never have YouTube to client traffic going to Chicago PoP under the location based load balancing. *Hence, location based load balancing will have a positive impact on ISP-X's PoP level traffic matrix, since it reduces the traffic on ISP-X's backbone network significantly.*

Poor resource utilization under geographical load-balancing. Although geographical load-balancing helps in reducing the amount of traffic carried in the *ISP-X*'s backbone, it results in poorer resource utilization and increased peak load at each data center.

To analyze the impact of location based load balancing on YouTube data centers, we compute the amount of video requests served by each data center during different time of the day. We estimate the amount of video requests served by each data center by rearranging the client to YouTube data center matrix on the basis of nearest PoP. In Fig. 15, we compare the loads at six data centers under location based and resource based load balancing schemes. In each figure here, x-axis represents the time and y-axis represents the total number of flows received by the corresponding data center at different time. We observe that the load on New York and Dallas data center increases significantly as they are the nearest data centers for a large number of *ISP-X* PoPs.

Furthermore, let us assume that the maximum number of flows each data center receives in any hour indicates the load at that data center. Let L_i^j indicate the number of flows received by data center i in hour j, then we define the load $L_{max}(i), i = 1, 2, \ldots, 6$ as, $L_{max}(i) = \max_{j=1 \ to \ 72} L_i^j$. Under these assumptions, the sum of peak load at each data center ($ML = \sum_{i=1}^{6} L_{max}(i)$) will represent the amount of computing resources needed during the peak load time to serve all the client requests. If we compare MLs under the two schemes, we see that the total units of resources that are needed in case of location based load balancing is about 30% higher than that of the current resource based load balancing.

The reason for this increase is the following. Since the peak load time for client requests coming from different PoP is different, the peak load at one of the PoPs of *ISP-X* is proportionally shared by all the data centers in the location agnostic load balancing. However, in case of location based load balancing it does not allow the distribution of peak

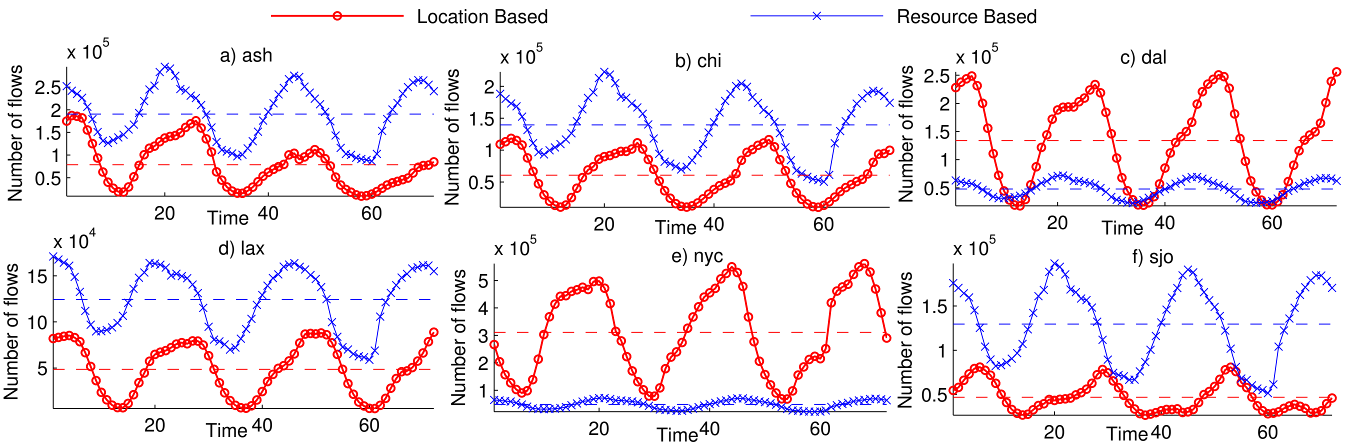


Figure 15: Comparison of traffic at different YouTube data centers for resource based and location based traffic distribution schemes.

load, and all the load accumulates at the nearest YouTube data center.

In summary, we see that while location based load balancing may reduce the load on *ISP-X*'s backbone network, it results in the poor resource utilization and creates more "hot-spots" for YouTube by increasing the load on the individual data centers during the peak load hours.

• **Scenario 2: What if YouTube only announces IP prefixes specific to each data center at the corresponding location where it exchanges traffic with *ISP-X*?**

Since YouTube announces larger all-inclusive prefixes at each location where it exchanges the traffic with *ISP-X*, the ISP can do early exit routing and dump all YouTube-bound traffic to YouTube at the nearest exit (see Sec. 5). This forces YouTube to carry that traffic internally to its different data centers. In this section, we investigate how traffic matrices will change if YouTube only announces specific prefixes from each of its location.

To see how the traffic matrix for *ISP-X* changes if YouTube starts announcing only the specific local prefixes at each location, we divided YouTube IP addresses in smaller prefixes so that each prefix has IP addresses belonging to only one YouTube data center. We combined smaller prefixes to form larger prefixes as long as the larger prefixes included IP addresses from a single data center. From the three large prefixes we obtained 2 /25s, 9 /24s and 5 /23s where each of the prefixes belonged to exactly one data center.

If YouTube announces only the specific prefixes from each location corresponding to the local data center, then *ISP-X* will have to carry the traffic corresponding to each YouTube prefix to the data center where it is present. This means most of client to YouTube server traffic will be carried in the ISP backbone.

However, as we have already seen in Sec. 4.5, the ratio between client-to-YouTube and YouTube to client traffic is approximately $1:59$, the additional volume of traffic that is carried in *ISP-X* backbone will only increase by 2% for the total YouTube traffic.

• **Scenario 3: What if due to BGP routing changes the number of prefixes YouTube reaches using *ISP-X* increases?**

As seen in the Sec. 6, a significant portion of the traffic the customers of *ISP-X* exchange with YouTube is not visible to *ISP-X*. They are likely being carried by some other ISPs. If YouTube changes its routing policy and decides to send all that traffic through *ISP-X*, the amount of YouTube to client traffic handled by *ISP-X* would increase significantly: by about 50% of the current YouTube to client traffic volume. This might also happen if YouTube experiences routing failures in its other providers' network. Since YouTubetraffic amounts for about 5.8%, this results in about 2.9% increase in the total traffic carried by the ISP. On the other hand, being able to estimate this unseen traffic accurately, *ISP-X* can prepare and plan for such scenarios in advance.

8. CONCLUSIONS

In this paper, we presented the first extensive study of traffic exchanged between YouTube data centers and users, as seen *from the perspective of a large tier-1 ISP*. Using our trace-driven analysis of the interaction between YouTube and one of its large transit-providers, we inferred and analyzed the load balancing scheme employed by YouTube and how these choices affect the traffic dynamics across a large content provider and a tier-1 ISP. We further formulated and solved unseen traffic estimation problem that ISPs can use to get an estimate of the "new" traffic demand and prepare for the potential impact on its network, when routing changes lead their to re-route unseen traffic through the ISP instead. We also conducted several "what if" analysis to understand how the load balancing strategies chosen by content providers and routing policies used by the ISPs affect each other.

Although this paper focuses primarily on the traffic characteristics of YouTube, the methods used have general applicability. In fact, similar experiments can be easily carried out for any other content-providers to understand how their load-balancing and routing choices affect their and their ISPs routing matrices.

9. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and our shepherd, Walter Willinger, for their helpful feedback on this paper.

We gratefully acknowledge the support of our sponsors. The work is supported in part by the NSF grants CNS-0626812, CNS-0721510, CNS-0905037, CNS-1017647 and the DTRA Grant HDTRA1-09-1-0050. The opinions expressed in this paper are solely those of the authors and do not necessarily reflect the opinions of any of the funding agencies or the University of Minnesota.

10. REFERENCES

[1] Z. Al-Qudah, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van der Merwe. Anycast-aware transport for content delivery networks. In *WWW '09*, pages 301–310, New York, NY, USA, 2009. ACM.

[2] T. Brisco. DNS Support for Load Balancing. RFC 1794, April 1995.

[3] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC '07*. ACM, 2007.

[4] X. Cheng, C. Dale, and J. Liu. Statistics and social network of youtube videos. In *Proc. of IEEE IWQoS*, 2008.

[5] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. *IEEE Internet Computing*, pages 50–58, 2002.

[6] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Youtube traffic characterization: a view from the edge. In *IMC '07*. ACM, 2007.

[7] C. Huang, A. Wang, J. Li, and K. W. Ross. Measuring and evaluating large-scale cdns (paper withdrawn). In *IMC '08*, pages 15–29, New York, NY, USA, 2008. ACM.

[8] V. Jacobson, C. Leres, and S. McCanne. Tcpdump Man Page. *URL http://www.tcpdump.org/tcpdump_man. html*, 2003.

[9] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving beyond end-to-end path information to optimize cdn performance. In *IMC '09*, pages 190–201, New York, NY, USA, 2009. ACM.

[10] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. *ACM SIGCOMM Computer Communication Review*, 32(4):174, 2002.

[11] W. Norton. The evolution of the US Internet peering ecosystem. Equinix White Papers, 2004.

[12] M. Saxena, U. Sharan, and S. Fahmy. Analyzing video services in web 2.0: a global perspective. In *NOSSDAV '08: Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 39–44, New York, NY, USA, 2008. ACM.

[13] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. *SIGMETRICS Perform. Eval. Rev.*, 32(1):307–319, 2004.

[14] A. Vakali and G. Pallis. Content delivery networks: Status and trends. *IEEE Internet Computing*, 7(6):68–74, 2003.

[15] T. Will. Introduction to the Singular Value Decomposition. *La Crosse, WI*, 2003, 2003.

[16] M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of youtube network traffic at a campus network - measurements, models, and implications. *Comput. Netw.*, 53(4), 2009.

APPENDIX

A. CHANGES AFTER WE COLLECTED THE DATA

After this data was collected in Spring 2008, we have observed several ways in which YouTube video delivery has changed. Most importantly, we have seen that YouTube video delivery system now uses IPs that belong to Google (AS15169) and even the DNS names for the video hosting servers are using google.com domain. For example, `www.youtube.com` now has a CNAME record mapping it to `youtube-ui.l.google.com` which is then mapped to different IP addresses when accessed from different geographic locations. Since we do not have access to any new Netflow data, we did some active measurements. We tried to access `www.youtube.com` from hundreds of Planet-Lab nodes and based upon ICMP ping latency and traceroutes to the IPs the DNS servers return for `www.youtube.com`, we learned that IPs mapping to `www.youtube.com` are part of different geographic locations. Therefore, we conjecture that the traffic to the YouTube front-end servers is most likely being served by a data-center which is geographically closer to the user. We do not have any new information regarding the video traffic, other than the facts that (a) the video servers IPs have changed and most of video server IPs are coming from IP prefixes belonging to Google and (b) the packet size distribution shown in Fig. 2 still holds true.