Improving Content Delivery Using Provider-aided Distance Information

Ingmar Poese
T-Labs/TU Berlin
ingmar@net.t-labs.tu-berlin.de

Benjamin Frank
T-Labs/TU Berlin
bfrank@net.t-labs.tu-berlin.de

Bernhard Ager
T-Labs/TU Berlin
bernhard@net.t-labs.tu-berlin.de

Georgios Smaragdakis
T-Labs/TU Berlin
georgios@net.t-labs.tu-berlin.de

Anja Feldmann
T-Labs/TU Berlin
anja@net.t-labs.tu-berlin.de

ABSTRACT

Content delivery systems constitute a major portion of today's Internet traffic. While they are a good source of revenue for Internet Service Providers (ISPs), the huge volume of content delivery traffic also poses a significant burden and traffic engineering challenge for the ISP. The difficulty is due to the immense volume of transfers, while the traffic engineering challenge stems from the fact that most content delivery systems themselves utilize a distributed infrastructure. They perform their own traffic flow optimization and realize this using the DNS system. While content delivery systems may, to some extent, consider the user's performance within their optimization criteria, they currently have no incentive to consider any of the ISP's constraints. As a consequence, the ISP has "lost control" over a major part of its traffic. To overcome this impairment, we propose a solution where the ISP offers a Provider-aided Distance Information System (PaDIS). PaDIS uses information available only to the ISP to rank any client-host pair based on distance information, such as delay, bandwidth or number of hops.

In this paper we show that the applicability of the system is significant. More than 70% of the HTTP traffic of a major European ISP can be accessed via multiple different locations. Moreover, we show that deploying PaDIS is not only beneficial to ISPs, but also to users. Experiments with different content providers show that improvements in download times of up to a factor of four are possible. Furthermore, we describe a high performance implementation of PaDIS and show how it can be deployed within an ISP.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Client/Server; C.2.5 [**Local and Wide-Area Networks**]: Internet

General Terms

Measurement, Performance

Keywords

Content Distribution, Host Diversity, Server Selection, DNS Redirection, Residential Traces

IMC'10, November 1–3, 2010, Melbourne, Australia.

1. INTRODUCTION

The Internet has evolved into a system where users can easily share content with friends and/or other users via applications such as online social networks, video portals, One-Click Hosters, Web services, wikis, blogs, or P2P file-sharing applications. In terms of volume, multi-media content, including photos, music, and videos, as well as software downloads and updates, are major contributors and together responsible for most of the Internet traffic [24, 22, 31, 29]. Indeed, HTTP is used to access this information and therefore accounts for more than 50 % of the traffic [3, 22, 24, 31, 29]. Moreover, it is hardly (mis-)used as a transport protocol for other applications [24]. Among the causes for the increase of HTTP traffic are the increase of streaming content, e.g., offered by `youtube.com` and the popularity of the content offered by One-Click Hosters [7] such as `rapidshare.com` or `uploaded.to`.

This content is hosted by the new "Hyper Giants" [22] which include large content providers, such as Google and Yahoo, as well as Content Distribution Networks (CDNs), like Akamai and Limelight. Most of these Hyper Giants are operating not only a substantial number of data centers but are also building up their own network [21]. Some networking researchers are claiming that, due to the phenomenal growth of Hyper Giants, the topological structure of the Internet must be redrawn to include them, together with the Global transit and backbone networks as part of the Internet core, resulting in the topology sketched in Figure 1. This may leave the ISPs as dump pipe providers to the consumer.

To achieve high levels of performance and scalability, most content delivery architectures, including well provisioned hosting and content distribution networks, heavily rely on a distributed infrastructure. Indeed, some of them have deployed hosts in more than 5000 locations throughout the Internet [23]. What's more, the content delivery providers have full control of their transfer process. Therefore, they can optimize the traffic flows so that it minimizes operational costs as long as quality of service agreements with the content producers are met. With the help of the DNS system, they can redirect requests to hosts within their infrastructure and minimize their operational costs. This may result in sub-optimal content delivery performance to the end-users, while imposing a heavy burden on the ISP caused by the pure traffic volume of the content as well as the control loop interactions. If the ISP changes its routing, e.g., for the purpose of traffic engineering, the content delivery network may re-optimize this delivery strategy and change the traffic matrix, which may render the traffic engineering choice of the ISP void [18]. Even though we show in Section 2 that most content is available at multiple locations throughout the Internet, the ISP has no choice regarding where users fetch content from. This decision is, at this point, left to the content delivery network, even though

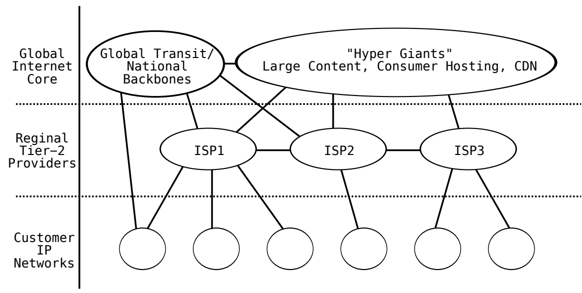

Figure 1: The new Internet Reality [22]?

the content delivery architectures provide a diverse set of hosts for any requested object.

Nevertheless, the ISP has substantial information at its fingertips which can be used to improve overall traffic flow and thus lower network utilization. This, in turn, can increase content delivery performance. However, this information is not readily available to the CDN. Rather, the CDN has to infer such information either via active measurements or client reports. Therefore, we claim that it is possible to improve content delivery using Provider-Aided Distance Information (PaDIS), see Section 3, much in the same way as P2P content delivery can be improved using ISP-P2P collaboration.

Previous work, e.g., [8][19][28] has in principle shown potential for biasing location choices. Indeed, we proposed that each ISP offers a service to the P2P users which explicitly helps them to choose "good" neighbors by ranking possible peers [5, 4]. Such traffic localization mechanism for P2P traffic, as proposed by Aggarwal et al. [5, 4], Xie et al. [36], Choffnes and Bustamante [11], are currently under discussion within the IETF ALTO [32] working group.

Network based information is in particular useful to circumvent bottlenecks, or to handle application flash crowds and other moving targets. Moreover, if an ISP has more control over its traffic flows, it can choose servers more intelligently and thus avoid network bottlenecks or choose closer servers in terms of network hops. This can both reduce overall traffic volume, as well as network utilization, which may further reduce network based congestion and thus improve the overall user performance.

We therefore propose to improve content delivery using PaDIS. Using active measurements, we show that choosing the appropriate server for content delivery can improve the user's download experience by up to a factor of four; see Section 4.

Our contributions in this paper are:

- To show that more than half of the total traffic, including the dominant HTTP traffic, can be delivered in principle from multiple servers at diverse network locations based on observations from passive packet-level monitoring of more than 20,000 residential DSL lines from a major European ISP.
- To capitalize on the diversity of content delivery, we propose to deploy PaDIS as a content location recommendation system, within the ISP. PaDIS can be interfaced with the ISP's DNS to redirect traffic, or it can be contacted directly by the client system.
- To quantify the content delivery performance improvement when using PaDIS in the wild, within a major European ISP. We report on our experience of downloading content from content distribution networks, and show significant potential for performance improvements. More specifically, we report on our experience when downloading content from different content distribution networks as well as One-Click Hosters. Our results clearly show that the end-user download time is significantly improved, while the ISP regains the ability to perform traffic engineering by biasing application layer choices.

To the best of our knowledge, this is the first work that proposes application and ISP collaboration based on the observation that content is usually accessible at multiple locations due to the prevalence of distributed content delivery architectures. However, there is a chance that the content delivery architecture may try to boycott the choice of the ISP by no longer exposing the multiple locations. However, since the ISPs control the access to the eyeballs, they can still use the principle approach as a negotiation tool. Moreover, with the help of the information from the ISP, it is possible to improve user performance substantially.

The remainder of the paper is structured as follows: In Section 2, we provide evidence in support of the potential diversity of hosts from which content can be downloaded. In Section 3, we provide the architecture of PaDIS and we show how it can be deployed within an ISP. In Section 4, we quantify the content delivery performance improvement while using PaDIS in the wild. We put our work in context with previous related research in Section 5 and conclude in Section 6.

2. CONTENT SERVER DIVERSITY

To highlight that a significant amount of traffic can, in principle, be delivered from multiple different servers at diverse network locations, we rely on passive network traces to identify popular services and active measurements to identify server location diversity.

2.1 Residential ISP Traces

We base our study on three sets of anonymized packet-level observations of residential DSL connections collected at aggregation points within a large European ISP. Our monitor, using Endace monitoring cards, allows us to observe the traffic of more than 20,000 DSL lines to the Internet. The data anonymization, classification, as well as application protocol specific header extraction and anonymization is performed immediately on the secured measurement infrastructure using the Bro NIDS [27] with dynamic protocol detection (DPD) [15].

We use an anonymized 24 h packet trace collected in March 2010 (MAR10) for detailed analysis of the protocol behavior. For studying longer term trends, we used Bro's online analysis capabilities to collect an anonymized protocol specific trace summary (HTTP-14d) spanning 2 weeks. Additionally, we collected an anonymized 5 day DNS trace (DNS-5d) in February 2010 to achieve a better understanding of how hostnames are resolved by different sites. Due to the amount of traffic at our vantage point and the resource intensive analysis, we gathered the online trace summaries one at a time. Table 1 summarizes the characteristics of the traces, including their start, duration, size, and protocol volume. It is not possible to determine the exact application mix for the protocol specific traces, as we only focus on the specific protocol. However, we use full traces to cross check the general application mix evolution.

2.1.1 *Popular Services*

With regards to the application mix, see Table 1, Maier et al. [24] find that HTTP, BitTorrent, and eDonkey each contribute a significant amount of traffic. In MAR10 HTTP alone contributes almost 60 % of the overall traffic at our vantage point, BitTorrent and eDonkey contribute more than 10 %. Similar protocol distributions have been observed at different times and at other locations of the same ISP. Moreover, these observations are consistent with

Table 1: Summaries of anonymized traces.

Name	Type	Start date		Dur	Size	Application Volume
MAR10	packet	Thu	04 Mar'10 2am	24 h	>5 TB	> 3 TB HTTP, > 5 GB DNS
HTTP-14d	log file	Wed	09 Sep'09 3am	14 d	> 200 GB	corresponds to > 40 TB HTTP
DNS-5d	packet	Wed	24 Feb'10 4pm	5 d	>25 GB	> 25 GB DNS

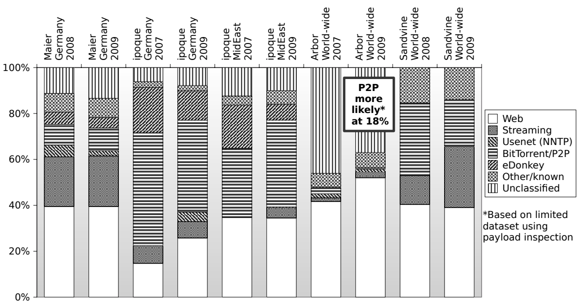

Figure 2: Barplot of Internet Application Mix (unified categories) across years and regions from multiple sources.

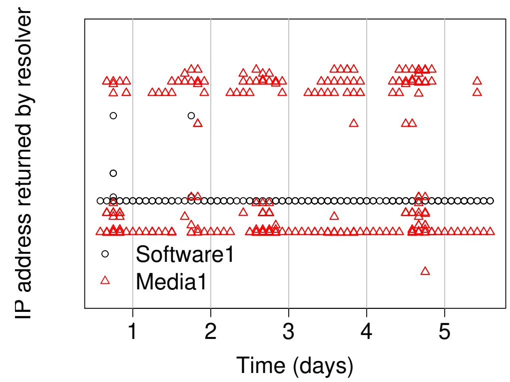

Figure 3: DNS replies for two different CDN-ized sites, in two-hour bins.

other recent Internet application mix studies [24, 22, 31, 29]. Figure 2 [30] summarizes the results of these studies. Note that almost all streaming is done via the Web on top of HTTP. Therefore, we conclude that HTTP is the dominant service.

Analyzing HTTP-14d, we find more than 1.2 billion HTTP requests, or 89 million requests per day on average. This is consistent with 95 million requests in 24 hours in MAR10. The advantage of using click stream data from a large set of residential users is their completeness. We are, e.g., not biased by the content offered *(i)* by a Web service, *(ii)* whether sufficient users installed measurement tools such as the `alexa.com` toolbar, or *(iii)* whether users actually use some kind of Web proxy.

To identify the most popular Web services, we focus on the most popular hosts. As expected, the distribution of host popularity by volume as well as by number of requests is highly skewed and is consistent with a Zipf-like distribution as observed in other studies [24]. The top 10,000 hosts by volume and the top 10,000 hosts by number of requests together result in roughly 17,500 hosts. This indicates that on the one hand, some hosts that are popular by volume may not be popular by number of requests and vice versa. On the other hand, there are some hosts that are popular according to both metrics. The total activity by these hosts accounts for 88.5 % of the overall HTTP volume and more than 84 % of the HTTP requests. Assuming that the HTTP traffic volume accounts for roughly 60 % of the total traffic, similar to the observations made in September 2009 [24, 3] and in MAR10, more than 50 % of the trace's total traffic is captured by these hosts.

2.2 Server Diversity and DNS Load Balancing

To better understand how HTTP requests are handled and assigned to servers, we use DNS-5d to analyze the 20 most heavily queried DNS names to identify typical usage patterns. We consider only the most heavily used resolver. Figure 3 shows two of the typical patterns for two of the DNS names. It also shows how the resolved IP addresses change (y-axis) across time (x-axis) for two hostnames; respectively a software site, labeled Software1, and a media site, labeled Media1. The vertical lines annotate midnight. If two IP addresses are plotted close to each other, this indicates that the longest common prefix of the two addresses is close. We note that the hostname of Software1 is mainly resolved to a single subnet, excepting a few special cases. However, Media1 is load balanced across approximately 16 different sites. For Media1, there appears to be one main site which is almost always available, while the remaining 15 are predominantly used during afternoon and evening peak usage hours.

These results are promising, and show that individual sites do expose a certain degree of server diversity to their users. While our trace (HTTP-14d) includes the queried hostnames, it does not include the resolved IP address, as a HTTP request header contains the hostname but not the IP address of a server. To verify the above behavior and get an up-to-date view of the DNS replies for the hostnames of our trace, we used 3 hosts within the ISP to issue DNS queries to the ISP's DNS resolver for all 17,500 hostnames repeatedly over a fourteen day measurement period starting on Tue Apr 13th 2010. During these two weeks, we received more than 16 million replies. Unless otherwise mentioned, we rely on our active DNS measurements, with augmented statistics concerning volume and requests from HTTP-14d.

2.3 Server Location Diversity

Our analysis of hostnames and their assignment to servers in section 2.2 has shown that content can be served by multiple servers in different locations. In fact, many domains use the service of a *Content Delivery Service* (CDS), which can be seen during the name resolution progress: The original domain name is mapped to the domain of a CDS, which then answers requests on behalf of the requested domain name from one of its caches [34]. Almost all CDSs rely on a distributed infrastructure to handle the expected load, load spikes, flash crowds, and special events. Additionally, this introduces needed redundancy and fail over configurations in their services. Among the most studied CDS' are Content Distribution Networks (CDNs), such as Akamai [23, 34, 17], and Content Delivery Platforms (CDPs), such as Google [21] and their YouTube service [10].

To better understand the DNS resolution process for hostnames hosted on CDS infrastructure, we refer to the machine requesting content as the `DNS client`. Along the same lines, we refer to

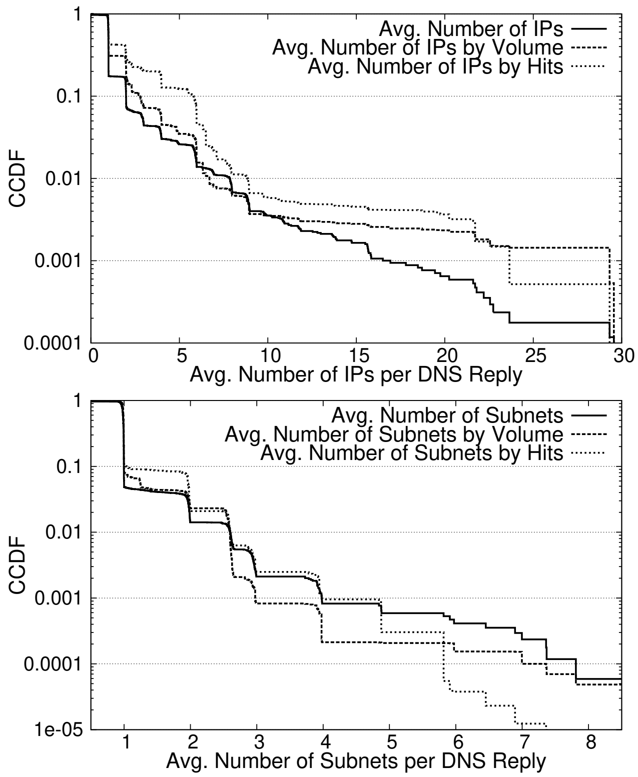

Figure 4: CCDF of mean # of IPs (top) and subnets (bottom) per DNS reply for the ISPs DNS resolver.

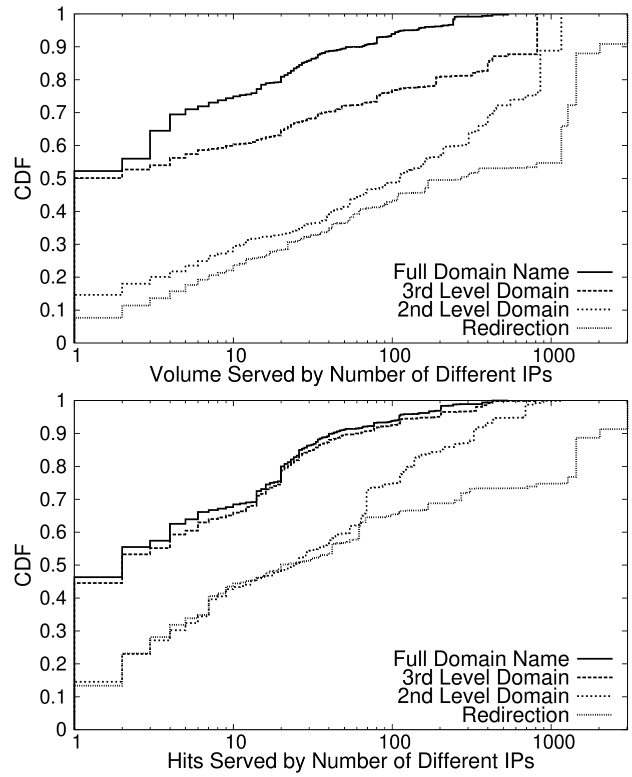

Figure 5: CDF of # of IPs for the ISP DNS resolver normalized by traffic volume (top) and requests (bottom) including aggregation on domain levels. (Logarithmic x-axis.)

the DNS server that receives the query from the client as the `DNS resolver`. This is usually run by the ISP or a third party DNS infrastructure like OpenDNS, also acting as a cache. Lastly, the authoritative DNS server, henceforth referred as `DNS server`, which is usually run by the CDS, replies to the DNS resolver. The DNS resolver caches the reply and hands it back to the DNS client.

The DNS server can choose to return one or more server IP addresses based on the domain name in the request and the IP address of the requesting DNS resolver. For example, it may use a geolocation database [33] to localize the region of the DNS resolver, utilize BGP data to identify the ISP, create a topology map derived via traceroutes, or any combination of these and other topological and geographic localization techniques. A DNS server has, in principle, two methods for load balancing across multiple servers:

MultQuery: Can return multiple IP addresses within a single DNS response

CrossQuery: Can return different IP addresses for repeated queries and thus perform DNS redirection.

In our active DNS measurements, we found that often a mixture of MultQuery and CrossQuery is being used in practice. Furthermore, we used the measurement results to *(i)* map hostnames to sets of IP addresses and *(ii)* check the IP address diversity of these sets for a better understanding of server diversity and their location. We achieved this by aggregating the returned IP addresses into subnets based on BGP information obtained from within the ISP. This allows for detailed information about the different locations within the ISP, while giving an aggregated view of subnets reachable via peering links.

Another issue stems from the fact that the IP address returned by the CDS depends on the IP address of the ISP DNS resolver [2, 26, 34]. Due to this, we used the DNS resolver of the ISP of our vantage point as well as external DNS resolvers (see section 2.3.1). The former reflects the experience of most of the clients at our vantage point[1]. The latter lets us discover additional diversity as well as understand the preference of the CDS for this specific ISP.

Prevalence of MultQuery.

We start our analysis by checking the prevalence of the first form of DNS based load balancing, MultQuery. Figure 4 shows a CCDF plot of the average number of IP addresses (top) and subnets (bottom) per DNS reply. In addition, we included the same data normalized by traffic volume and number of requests.

A first observation is that the number of returned IP addresses per request is rather small. The median is 1, the average is 1.3 and even the 0.9 percentile is 2. We note that even when an answer yields multiple IP addresses, the majority of them are from the same subnet. Therefore, the diversity decreases even further if we aggregate to subnets. From a network perspective, this implies that there is not much choice, neither for the ISP nor for the user, regarding where to download the content from. Both are limited to the information provided by the DNS server. However, when we normalize the hosts by their respective popularity, we see a significant improvement. More than 29% of the volume and 19% of requests have a choice among at least 2 IP addresses.

[1]We verify using the traces that more than 95 % of the clients use the ISP's DNS resolver as their default one.

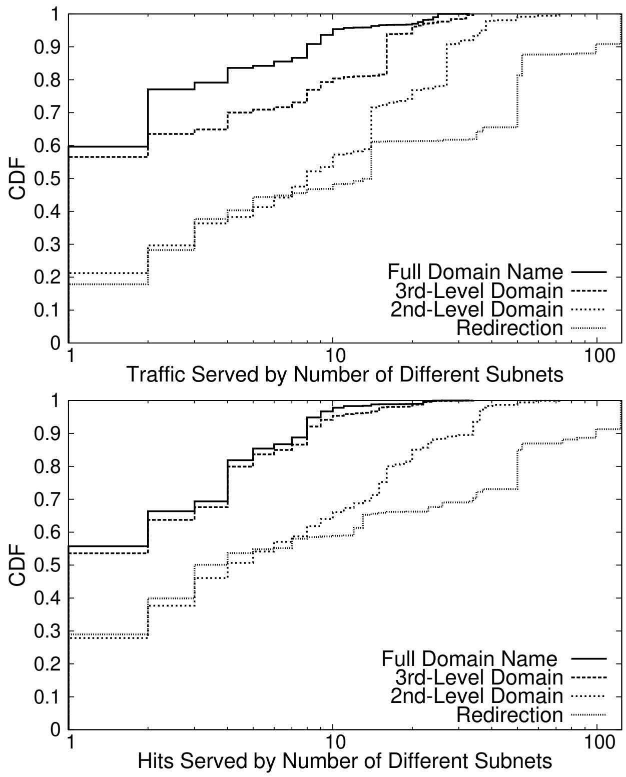

Figure 6: CDF of # of subnets for ISP DNS resolver normalized by traffic volume (top) and by requests (bottom) including aggregation on domain levels. (Logarithmic x-axis.)

Prevalence of CrossQuery.

Next, we check how prevalent CrossQuery, the second form of DNS based load balancing is. Since CrossQuery returns different IP addresses for repeated queries, its potential contribution to server diversity can only be studied by aggregating across time. The lines labeled `Full Domain Name` in Figures 5 and 6 capture this case.

We find that more than $50\,\%$ of the volume or requests can be served by more than one IP address. similarly, there is choice between at least two subnets over $40\,\%$ of the time across both metrics, see Figure 6. This indicates that there is significant potential for the ISP to bias the location preference of the CDS.

Subdomain Aggregation.

Since some CDSs only use subdomains as hints about the context of the requested URLs or the requested services, we accumulate the answers further regarding the 2nd and 3rd part of the domain names of the hosts, see Figures 5 and 6 at the respective data series called `3rd Level Domain` and `2nd Level Domain`. For example, we might accumulate the IP addresses from DNS replies for `dl1.example.org` and `dl2.example.org` for the statistics on the 2nd level domain, but not the third level domain.

This is a feasible approach, since many hosts respond to all requests that belong to a subset of the subnets returned when accumulating by the second-level domain of DNS resolver answer, including recursive requests and redirections. We verify this behavior with active measurements, see Section 4. We find that at least two major CDNs, a streaming provider and a One-Click Hoster, serve requested content from servers that match in their second level domain.

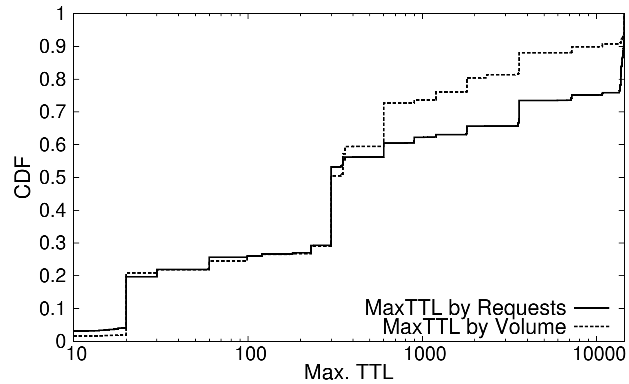

Figure 7: CDF of DNS TTL value by traffic volume and by number of requests.

We note that the accumulation by third-level domain, and especially by second level domain significantly increases the number of observed subnets per request both normalized by requests as well as by volume. The number of returned subnets further increases when accumulating to the second-level domain of DNS resolver answer. Studying our traces in more detail, we find that this is due to the substantial traffic volume and number of requests that are served by CDNs, some of which are highly distributed within ISPs or located in multihomed datacenters or peer-exchange points.

Infrastructure Redirection Aggregation.

Taking a closer look at the DNS replies [25], we find that some CDSs use CNAME records to map queried hostname to an A record. These A records show the same pattern as the hostnames in the previous section: the second level domain is identical. Similar to the previous approach, we can aggregated by these A records.

For example, at some point in time the hostname `www.bmw.de` is mapped via a CNAME chain to an A record with the name `a1926.b.akamai.net`, while `www.audi.de` is mapped to `a1845.ga.akamai.net`. Since the second level domain on the A records match, these DNS replies will be aggregated. Indeed, it has been shown that both caches will serve the content of either website [35]. On the down side, it is possible that this scheme of aggregation reduces the effectiveness of the CDN's caching strategy. This aggregation is called `Redirection` in Figures 5 and 6.

Turning our attention to the implications of the proposed aggregation schemes, we notice the available diversity increases tremendously. More than 50% of the hits and 70% of the bytes can be served by more than 20 servers. With regards to subnets, the diversity decreases slightly. Nevertheless, more than 5 subnets are available for 45% of the hits and 55% of the bytes.

If we consider aggregation periods in the order of tens of minutes, the numbers do not decrease by much. The reason that most of the diversity is observable even over these short aggregation time periods, is that the typical TTL, see Figure 7, is rather short with a mean of $2,100$ seconds and an median of 300 seconds normalized by volume. When weighted by requests, the mean is $4,100$ seconds and the median is 300 seconds.

2.3.1 *Alternative DNS Resolvers*

So far we have only considered the effect of content diversity when the ISP DNS resolver is used. To understand how much the

DNS load balancing deployed by a CDS is biased by the queried DNS resolver, we repeat the experiment from Section 2.2 using two other DNS resolvers. In particular, we pick the next most popular DNS resolvers found in our traces: GoogleDNS and OpenDNS. Both are third-party resolvers with a global footprint and utilize DNS anycast.

Comparing the results, we find that we attain more IP address diversity and subnet diversity when using the ISP DNS resolver. This is mainly due to the fact that CDSs select the supplied caches based on the source IP address of the querying DNS resolver. Since the CDSs are no longer able to map the request to the AS it originates from, but rather to AS the DNS resolver belongs to, the server selection by the CDS cannot optimize for the location of the DNS client.

2.3.2 Impact on Traffic Localization

Analyzing the three active DNS measurements from the ISP, OpenDNS as well as Google DNS resolver, we find that a significant part of the requests that could have been in principle served by sources within the ISP are directed towards servers that are outside of the ISP. However, before tackling this issue, we need to understand what fraction of the traffic may be served by IP addresses within the ISP's network and what fraction is served by IP addresses outside of the AS. To this end, we analyze each of the three active DNS traces separately. For each trace, we start by classifying all DNS replies regarding the `redirection` aggregation described in section 2.3 and account the volume (or hits) evenly to each of the IP addresses. Next, we classify the IP addresses in two groups - inside and outside of the ISP network. Table 2 summarizes the results of this aggregation regarding the traffic and hits that were kept inside the ISP's network in the columns labeled `observed`.

Turning to the results, we find that there is hardly any difference between those clients that use the external DNS resolvers. Of the returned IP addresses, less than 6 % are within the AS. When weighted by number of requests, this does not change much. However, when normalizing by volume, about 12 % of the traffic stays within the AS.

In contrast, clients that use the ISP's DNS resolver fare better: almost a quarter of the traffic volume is served from servers within the AS. Normalized by requests, we see a three fold increase, and normalized by hits or volume, roughly a two fold increase over using external DNS resolvers. Among the reasons for the "bad" performance of external DNS resolvers is that some CDSs may always return IP addresses outside the ISP, despite the fact that many of its servers are deployed within the ISP. This explains the substantial difference and highlights on the one hand the effectiveness of the CDS optimization, but also points out its limits. As such, it is not surprising that there are efforts under way within the IETF to include the source IP addresses of the DNS client in the DNS requests [12].

However, one can ask if the CDS utilizes the full potential of traffic localization. For this, we check the potential of traffic localization, by changing the volume (or hit) distribution from even to greedy. Thus, as soon as we observe at least one IP address inside the ISP's network, we count all traffic for the entire aggregation to be internal. Table 2 shows the results in the columns labeled `potential` for all three DNS traces.

Note the substantial differences. Our results indicate that a gain of more than a factor of two can be achieved. Furthermore, up to 50 % of the traffic can be delivered from servers within the ISP rather than only 23.4 %. This may not only in itself result in a substantial reduction of costs for the ISP, but it also points out the potential of our proposed approach. While the increase is noticeable for OpenDNS, it is nowhere near that of the ISP's DNS resolver. The potential benefit when relying on GoogleDNS is rather small. A deeper study on our results unveils that content served by highly distributed and redundant infrastructure can be localized the most.

2.4 From Server Diversity to Path Diversity

Next, we ask the question whether the substantial diversity of server locations actually translates to path diversity. For this purpose, we generate a routing topology of the ISP by using data from an IS-IS listener and a BGP listener. However, due to the asymmetry of routing, we have to explore both directions separately. With the same argumentation as in Section 2.3 we choose to aggregate using the `redirection` scheme for calculating path diversity. For the HTTP requests we can determine the path within the ISP using the routing topology. We find that roughly 65 % of the total HTTP requests can be forwarded along at least two different paths. Indeed, roughly 37 % of the HTTP requests can be forwarded along at least four different paths.

In addition, we can use the routing data to determine the paths of all content that is potentially available within the ISP's AS.[2] We find that there is significant path diversity. In some cases, a request can follow up to 20 unique different paths. Moreover, we see that around 70 % of the HTTP traffic volume and requests can be sent along at least two different paths.

2.5 Summary

We see that HTTP is again the dominant traffic source, while the prevalence of P2P traffic decreases. Since most CDSs rely on distributed infrastructure, we not only observe significant server location diversity but also significant path diversity for accessing HTTP based content. Indeed, there is the potential to bias roughly half of the overall traffic by redirecting queries to different content servers.

More precisely, we estimate that around 70 % of the HTTP traffic in a big European ISP can be redirected when taking advantage of the diversity due to MultQuery, CrossQuery and hostname aggregation. Furthermore, we show that current CDS optimizations that approximate the location of end-users based on the location of the local DNS resolvers are more effective than those based on the location of third-party resolvers. Finally, we show that the traffic localization potential within the above mentioned ISP is very high especially when the ISP DNS resolver is utilized.

3. PaDIS ARCHITECTURE AND DEPLOYMENT

Given that a substantial fraction of the overall traffic is available at multiple locations within the ISP and that there is significant path diversity, we now propose a system, named PaDIS, that lets the ISP take advantage of this diversity while improving content delivery to the user.

PaDIS is the abbreviation for *Provider-aided Distance Information System*. PaDIS' task is to act as a location recommendation service and it is operated by an ISP. More specifically, a user, a CDN, a CDP, a DNS resolver [6], or any other entity can query PaDIS by submitting a list of possible IP addresses and a source. Upon receiving such a list, PaDIS will rank the submitted IP addresses according to its metrics such as distance within the Internet topology, path capacity, path congestion, path delay, etc. To be able to issue such a ranking, PaDIS relies on ISP specific network information, e. g., the local topology as well as Internet wide information, e. g., BGP information.

[2] Augmenting the routing topology with flow information may allow us to extend this analysis to all content.

Table 2: **Traffic localization within the network by different DNS resolvers normalized by number of requests and traffic volume together with the potentially available fraction of localized traffic.**

	ISP DNS		OpenDNS		GoogleDNS	
Metric	observed	potential	observed	potential	observed	potential
IPs	12.3 %	24.2 %	5.8 %	16.0 %	6.0 %	9.7 %
requests	14.9 %	33.2 %	4.7 %	18.8 %	4.8 %	6.4 %
volume	23.4 %	50.0 %	12.0 %	27.7 %	12.3 %	13.4 %

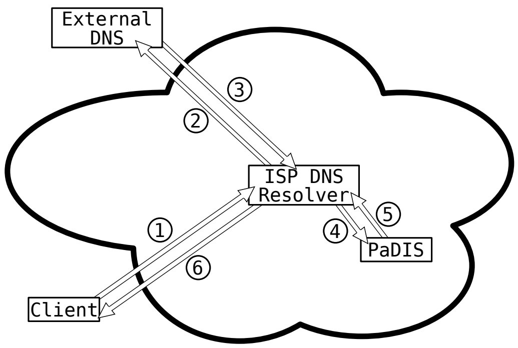

Figure 8: PaDIS use case: Optimizing content delivery transparent to both CDN/CDPs and clients.

We first outline how PaDIS can be used to take advantage of server diversity for content delivery, then outline its architecture, before discussing its scalability and responsiveness properties.

3.1 PaDIS Usage Options

While PaDIS can be used as an ALTO [32] server to offer ISP-aided localization for neighbor or peer selection for P2P users, it offers many more possibilities for optimizing content delivery. PaDIS rankings can either optimize for delay, e. g., for web sites where objects are typically small and the retrieval time is dominated by the round-trip-time (Section 4.1) or bandwidth, e. g., for One-Click Hosters (Section 4.2) offering bulk data.

Assuming that the local ISP runs a PaDIS server much in the same manner as it offers a DNS resolver to clients, CDSs, the ISP itself, etc. can use it in a multitude of different ways as outlined below:

Clients (a): Clients can install a plug-in to their Web browser to send all DNS replies or even summaries of past DNS replies to the PaDIS server for re-ranking the returned IP addresses taking the ISP's preferences into account.

Clients (b): Clients can overwrite the library responsible for DNS lookups with one that adds a PaDIS query and then re-ranks the DNS responses.

Clients (c): Another possibility is to pre-configure the home-router located at the edge of a client's network. Note, these routers also often act as DNS-relay. In this case, the home-router can also send any DNS reply to the PaDIS server and then return a re-ranked DNS reply.

CDNs/CDPs: Content delivery services may collaborate with the ISPs by contacting them before returning their server choices to the DNS resolver. A good heuristic for identifying an appropriate PaDIS server is to contact the ISP where the DNS resolver is located. This use case has the advantage that content delivery networks can take the ISP preferences into account during their optimization process. However, the CDS requires a hint as to the location of the client, e.g., its IP address. This is already under discussion within the IETF [12].

ISP: The ISP can enhance its DNS resolver to contact its PaDIS server and reorder the IP addresses if needed before returning any answer to a client. This is fully transparent to both the clients as well as the CDNs/CDPs. Figure 8 shows this scenario, which involves the following steps:

1. The client sends the DNS query to the ISP operated DNS resolver.
2. The query is recursively resolved using the authoritative DNS servers.
3. The reply is received by the ISP's DNS resolver.
4. The reply is sent to the PaDIS server for ranking.
5. The PaDIS server augments the reply with information from previous ones and ranks them according to its metrics, which takes the current network status into account. This reply is then sent back to the DNS resolver.
6. The ISP's DNS resolver sends the ranked and augmented reply back to the client.

One drawback to most of the above approaches is that most DNS responses are of the type CrossQuery and therefore do not contain a large number of possible server locations. However, as we have seen, DNS TTLs are usually short, so when aggregating across time, server diversity increases.

Let us revisit the DNS TTLs: Originally these were designed to ensure that stale cache entries do not linger forever in the DNS caches and that it is possible to relocate, add, or remove servers and domains within the DNS hierarchy. Today, however, DNS TTLs are significantly shorter than originally envisioned and are mainly used to aid CDSs with their load balancing and traffic flow optimization [34, 23]. Moreover, not all clients adhere to the DNS TTL values. Therefore, even today, a CDS has to add safety margins to the TTL values before they can stop serving specific content from a possible server. We propose to take advantage of this and allow the PaDIS server to keep a history of DNS hostname to IP address mappings. Using these mappings, the DNS replies can be augmented and clients can take full advantage of the server location and path diversity.

However, while the above ISP use case is transparent to the clients as well as the CDSs, we favor the CDN/CDP use case as it gives flexibility and control to both the CDNs/ CDPs as well as the ISP.

3.2 PaDIS Architecture

The designed architecture of PaDIS is shown in Figure 9. It has two main functionalities: to answer client ranking queries according to certain metrics, and to maintain a network information database. The latter includes network information, such as topology, routing, link capacity, link congestion, and link delay. It also needs to keep in sync with the network, e. g., the network has to be monitored for topology changes and changes to the path characteristics. Internally, PaDIS represents the network as an anno-

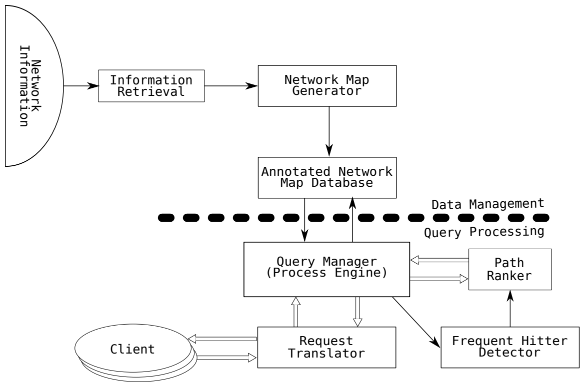

Figure 9: PaDIS architecture: Overview

tated graph. Accordingly, PaDIS is divided into two parts: data management and query processing, which communicate via shared memory.

Data Management Subsystem.

The data management subsystem consists of components for information retrieval, network map generation and the network map database.

The information retrieval component is responsible for continuously collecting network information, e. g., it listens to IGP messages, fetching status information and monitors routes learned via EGP. The ISP in question uses IS-IS as its IGP. The information retrieval component generates a representation of the physical network topology from the IGP messages it receives. In case of topology changes due to failures, addition of new components or scheduled downtime of monitored hardware, the topology is updated online. The information retrieval subsystem also fetches network status information and per link statistics, e. g., utilization and delay. In addition, the operator can assign customized values to links, e. g., related to operational costs or contract specifics.

Moreover, the information retrieval subsystem augments the ISP internal topology with external routing information by incorporating EGP messages. For the ISP in question these are BGP updates. Thus, by combining IGP and EGP information, PaDIS can find network path information, including performance characteristics, for any IP-based connection originating in the ISP. Customers of the ISP are handled in the same manner as EGP messages. However, their position and link characteristics may have to be learned from a data source such as a radius server. In short, the information retrieval subsystem learns as much as possible about the network topology of the ISP to generate a fully annotated network topology from it.

The network map generator is responsible for maintaining an up-to-date view of the topology supplied by the information retrieval subsystem. It pre-calculates the paths within the network and caches them for fast look-up. Since the routing within the ISP is typically more stable than customer and external prefixes, there is a significant benefit to caching the path, as it allows for a constant $O(1)$ look-up of the full path, regardless of the network's diameter, and a $O(1)$ complexity when updating the more volatile EGP information, e. g., BGP and/or radius information. However, recalculating the paths after a topology change costs $O(n^2)$ where n is the number of routers maintained by PaDIS.

Query Processing Subsystem.

The query processing subsystem consists of a request translator, a query manager, a path ranker and a frequent hitter detector.

The request translator component checks whether the query complies with the protocol specification and performs admission control based on the client IP address. Furthermore, PaDIS can be configured to augment requests with additional IP addresses to further enhance the choices presented to the client. If the request is admitted, the request translator reformats it and submits it to the Query Manager.

The query manager fetches all available information about each source-destination pair from the topology. Each pair, together with the path information, is then handed to the path ranker which uses a customized ranking function to calculate a weight representing the preference for this path. Once all pairs have been weighted, the query manager sorts the list by weights, striping all additional information added for ranking. The ordered list is passed back to the request translator and then returned to the client as a sorted list of sources in descending order of preference as seen by this ISP.

PaDIS is able to support a number of customized ranking functions. Each of these can optimize for different aspects of the path, e. g., bandwidth, hops, delay, etc., or a combination thereof. The client can specify its preference regarding the chosen metrics in the query, triggering the usage of an appropriate ranking function. However, the details of the functions are not revealed to the clients and the ranking weights are stripped from the reply. More importantly, no network information is revealed, contrary to other schemes of user-provider collaboration [36]. Note, that it is intractable to extract network properties or topologies from the ranking even when combined with traceroute information [1].

No information about clients is stored within the system, thus it preserves client privacy. However, to prevent abuse, PaDIS includes a frequent heavy hitter detector which can be activated by the ISP operator. The heavy hitter detector in our prototype is based on probabilistic lossy counting [13]. It maintains a list of the most popular IP addresses and sources which can also be utilized by the ranking function to avoid the creation of hot spots.

3.3 Scalability and Responsiveness

To decrease protocol handling overhead, we use UDP as the default protocol. However, TCP is supported for requests that exceed one MTU. Next we quantify the performance and scalability of PaDIS.

We use two identical dual quad-core CPU machines with sixteen gigabytes of memory directly connected via Gigabit Ethernet, one as PaDIS client and one as PaDIS server. We start by sending queries at a slow rate and gradually increase the pace until the PaDIS server is fully utilizing all CPUs, all the while ensuring that the client still receives all replies. Figure 10 shows the resulting number of served queries as well as the average delay for an eight-thread instance of the PaDIS server. Since the overhead depends on the number of IP addresses within the query, we varied this number from 50 to 363 — the maximum number of IP addresses for UDP-based queries which are restricted to one MTU. PaDIS is able to serve roughly 90,000 requests per second while ranking 50 IP addresses. This number drops to about 15,000 per second when ranking 363 IP addresses. Therefore, we conclude that a single PaDIS server offers sufficient throughput to be matched with one DNS resolver of the ISP.

The response time of the PaDIS server ranges from 1 millisecond to a few milliseconds and is dominated by the processing overhead in the hardware rather than the network connection. Moreover,

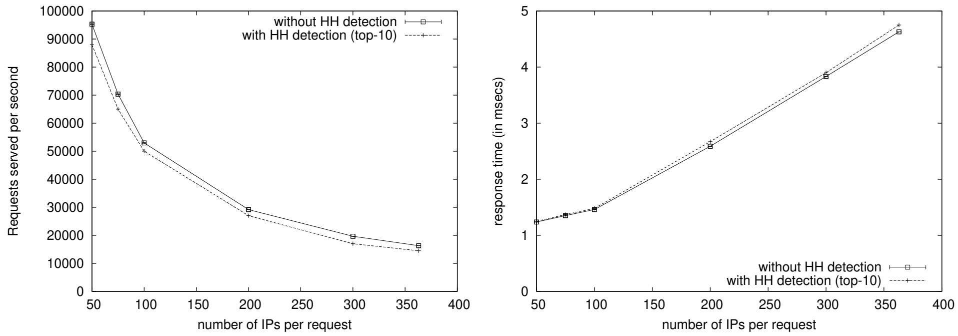

Figure 10: PaDIS performance: # of requests (left) average response time (right) with increasing query sizes.

CDN1		CDN2	
object#	size	file#	size
01	38K	01	36K
02	66K		
03	154K		
04	217K	02	254K
05	385K	03	471K
06	510K	04	599K
07	905K		
08	2.48M	05	3.4M
09	6.16M	06	4.5M
10	17M	07	8.6M

Table 3: CDN performance evaluation: Object sizes.

response time variance was minimal and the software ran stably throughout all experiments.

We repeated the experiment with heavy hitter detection activated and maintained a list of heavy hitters by considering the top ten replies, aggregating them every ten seconds. These were then fed into the ranking function. Our results show that both the number of requests served by PaDIS as well as the average response time decreases only slightly.

4. PERFORMANCE EVALUATION

To highlight that CDSs do not necessarily optimize their DNS load balancing strategies in such a way as to maximize user performance, and to show the potential of application ISP collaboration, which PaDIS enables, we perform extensive active measurements. Using ten vantage points within the ISP at residential locations and selected Web services that are responsible for a significant fraction of the HTTP traffic, we show that the server location diversity leads to different service performance results, and that PaDIS can help realize these. Among the studied Web services are the leading content providers, including the two most popular CDNs, the most popular One-Click Hoster (OCH) and the most popular streaming video provider.

The ten vantage points are deployed within residential locations with DSL connectivity to the ISP. The downstream bandwidth ranges from 1 Mbps to 25 Mbps while the upstream ranges from 0.128 Mbps to 5 Mbps. The measurements started on 1st of May 2010 and lasted for 14 days. Each client accesses the selected services 24 times during each day. In addition, we perform a DNS query for the hostname, in order to determine which IP addresses the service recommends. This methodology allows us to understand the possible end-user performance improvements. Moreover, we can estimate the network distances and thus the network savings. In the following section we show a selected subset of these measurements which represent the entire dataset collected.

4.1 Content Delivery Networks

Using the data sets from the residential ISP, see section 2.1, we identify the two most popular CDNs, referred to as CDN1 and CDN2. These are responsible for roughly 20 % of all HTTP traffic. Using the methodology discussed in Section 2, we identify more than 3,500 unique IP addresses that are caches for CDN1 and more than 700 unique IP addresses for CDN2. Both of these CDNs have more than 300 of their cache IP addresses within the ISP. In addition to CDN services, CDN1 offers an extensive range of service to content producers, including load-balancing and DNS services. If a content producer uses these services, it effectively outsources the DNS load balancing for requests to the CDN. However, the content is still provided by the content producer.

After augmenting each identified CDN IP address with its network path information, see Section 2, we find that the cache diversity translates not only into cache subnet diversity, but also path diversity. Thus, PaDIS can in principle be used to take advantage of both the server and the path diversity. However, at this point, it is still unclear whether the optimization of the CDN can be improved by PaDIS in terms of client download time and/or number of hops traversed within the ISP.

Since recent studies of CDN behavior have shown that any CDNized object is accessible from an arbitrary cache [17, 35], we can bypass the CDN recommendation. Thus, we request the URL directly from each of the identified CDN cache IP addresses regardless of their location. We verify this for all caches of CDN1. In addition, we point out that CDN1 caches also serve content of domains which only use their load balancing or DNS service in the same manner as if the content was supposed to be delivered by CDN1.

However, CDN2 is more restrictive. Our measurements show that CDN2 caches only reply to requests from the same region. In our case, we observe that European caches do serve the content to our European clients. However, when these requested content from North American caches, the delivery was refused.

Since the download performance of Web pages may depend on the size of the object, we select ten different files for CDN1 and seven for CDN2 of different but comparable file sizes ranging from 36 KB to 17 MB, see Table 3. To be able to repeat the measurements multiple times during a small time period while not overwhelming the client DSL lines, we subsample the number of caches of both CDNs. To preserve path diversity, we randomly select one cache from each of CDN1's subnets. This reduces the number of caches to 124 for CDN1. For CDN2 we found five subnets[3] containing caches, yet only two answered our queries with the requested data, as we have already explained. Since we use the raw IP addresses for accessing the various caches and override the CDN's server selection, we also exclude the domain name resolution time for the CDN recommended download.

Figures 11 and 12 show boxplots of the object retrieval time on a typical day (May 12^{th}) of our selected caches across time for CDN1 and CDN2 respectively, for a subset of the objects and one specific client. Inspecting the results from the other clients, and for the other objects, throughout the experiment we see similar results.

[3]Apparently, CDN2 utilizes well provisioned and well connected data centers around the world and thus relies on the redundancy within the data centers and their access to multiple ISPs.

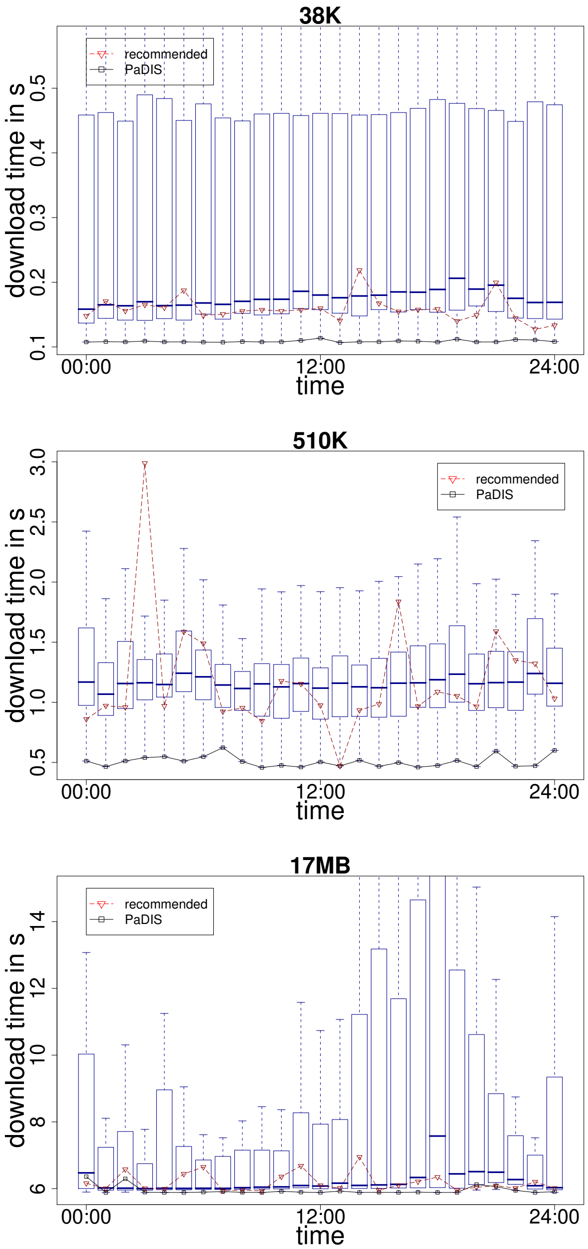

Figure 11: CDN1 performance evaluation: Boxplot of file downloads for the caches across time for objects 01, 06, and 10.

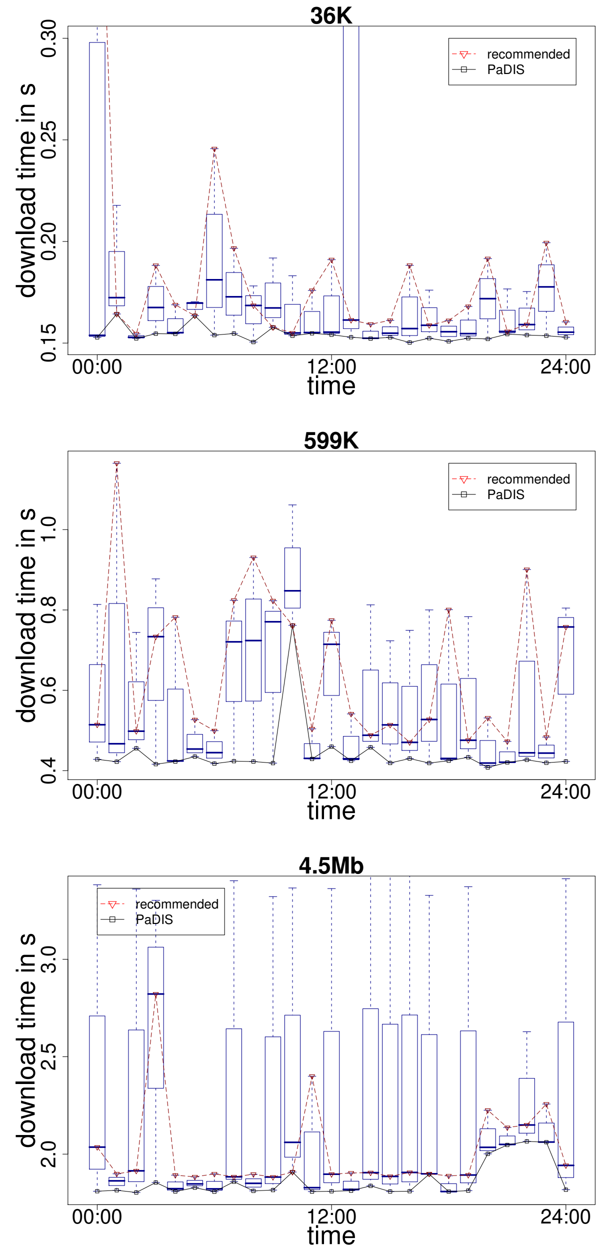

Figure 12: CDN2 performance evaluation: Boxplot of file downloads for the caches across time for objects 01, 04, and 06.

We use box plots because they are a good exploratory tool allowing the visual inspection of typical value ranges, spread, skewness, as well as outliers. Each Box analyzes the results of downloading the selected file at one point in time from one server in each of the subnets, e. g., for CDN1 each box consists of 124 data-points. The Box itself stretches from the 25th to the 75th percentile. The line within the box corresponds to the 50th percentile (the median). The whiskers represent the lowest and highest datum still within 1.5 times the interquartile range of the lower and upper quartile respectively. The dashed lines with triangles corresponds to the object download time for the recommended cache by the CDN. The solid line with squares corresponds to the object download time for the cache that ranked the highest by PaDIS based on delay.

A first observation regarding Figures 11 and 12 is that the download time for the recommended caches by CDN1 and CDN2 are quite good and close to the median download time over all the caches examined. Still, there is significant space from improvement especially during peak hours. Overall, PaDIS is able to improve the download time up to a factor of four.

Our active measurements also highlight typical network effects. For example, when downloading small objects, TCP is typically stuck in slow start. Thus, the round-trip time to the cache is the dominant factor for the retrieval time.

When downloading medium-size objects, both bandwidth and delay matter. For large objects the performance is restricted by the available network bandwidth including the download bandwidth

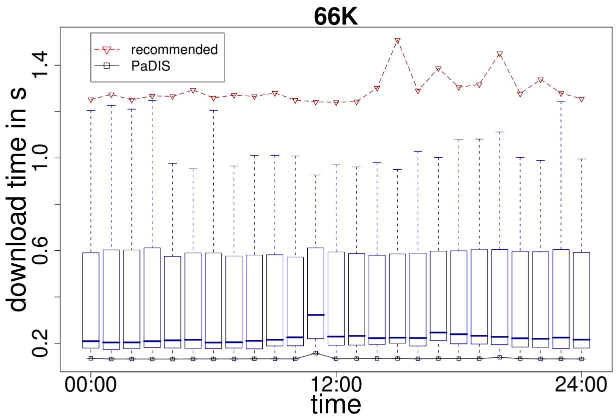

Figure 13: CDN1 load balancer performance evaluation: Boxplot of file downloads across time for object 03.

of the last-hop to the client (25Mbit/s in this experiment). For CDN1 the download time improvement for large objects is less than for small and medium ones, especially during the night, since the achieved download speeds are close to the nominal speed of the vantage points.

Since CDN1 also offers a pure DNS load balancing service, we are interested in examining if PaDIS can be helpful in such a scenario. Thus, we we repeat the experiment of fetching content from CDN1 recommended server as well as from all caches we associated with CDN1. As discussed before, the content of the website is served by all caches from CDN1. Also, CDN1 consistently returns the original server and never any of its own caches. In Figure 13 we plot the download time for objects that are retrieved using the above mentioned DNS load balancing service when following the recommendation by CDN1 and PaDIS. We also use boxplots to summarize the download time for the above mentioned 124 caches. The performance gain is substantial but has to be taken with caution as the recommended server is the original one and no CDN optimization takes place for this content. Nevertheless, we were able to use the CDN infrastructure to improve download time for content that was not distributed by the CDN. We limit the duration of our experiment, as such a behavior may violate the agreement between CDN1 and the site operator.

With respect to the ISP's benefits, we point out that PaDIS is able to localize the content within the ISP and that the average path length within the AS was reduced from 3.8 to 3 when downloading content from CDN1. Due to the small diversity of choices in CDN2, the internal path-length remained unchanged, even when the PaDIS decreased the download time.

4.2 One-Click Hosters

One-Click Hosters (OCH) offer users the ability to share files via a server based infrastructure, typically located within one or several well-provisioned data centers. Recent studies have shown that OCHs can achieve better download time than, e. g., P2P systems such as BitTorrent [7]. Therefore, it is believed that such services may become the leading platform for file sharing and replace P2P systems. Using our data sets from the residential ISP, we identify the most popular OCH, referred to as OCH1, which is responsible for roughly 15 % of all HTTP traffic. OCH1 is located at a multi-homed data center in central Europe. To scale the number of parallel flows, OCH1, like other OCHs, limits the maximum file size to 200 MByte.

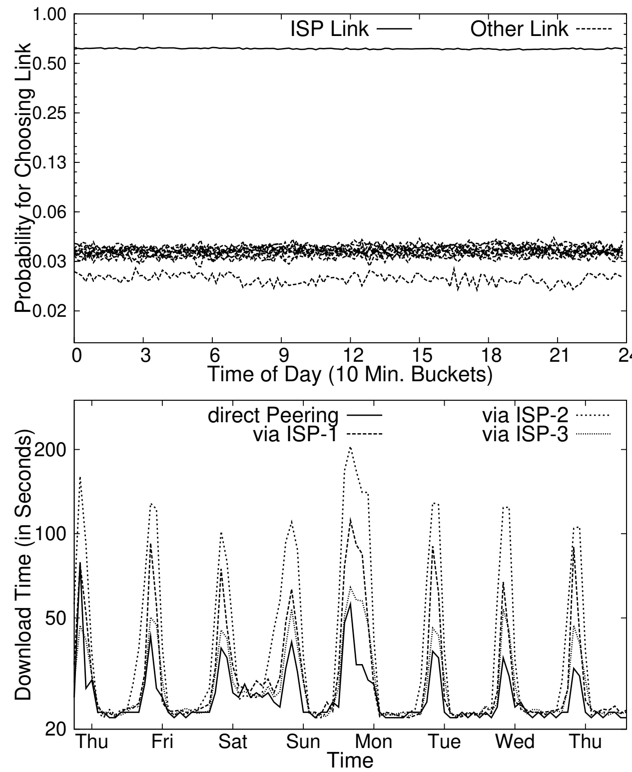

Figure 14: Probability of selecting a link of OCH1 (top) and download time per OCH1 link (bottom).

Using the traces, uploading a 60 Mbyte test file, as well as studying the domain name scheme of the servers, we are able to deduce that OCH1 has twelve uplinks to four different providers. The ISP we are collaborating with is among these providers. To understand how OCH1 does uplink selection, we repeatedly asked the OCH1 for a server to download the file during the one week period starting on the 7th of April 2010. The results, in terms of link selection probabilities for the twelve uplinks, are shown in Figure 14 (top). Roughly 60 % of the requests are shown to be directed to a server that can serve the content via the direct peering with the client's ISP. From the other eleven uplinks, ten uplinks are chosen with equal probability while one is chosen with smaller probability. It is worth noting that there are no time-of-day or time-of-week effects at all, while the HTTP volume of OCH1 in our traces exhibits time-of-day effects. This leads us to believe that the link utilization can be improved by using PaDIS' recommendation when assigning clients to servers.

From the replies of the OCH1, we derive a list of their available servers on a per uplink and provider basis. We validated that it is feasible to download the file from any one of the servers and thus download the content via any of the providers. To quantify the potential benefit, in download time, for the end-user, we repeatedly download the test file from one server per uplink once every two hours for one week. Figure 14 (bottom) shows the resulting download time for the client. For presentation purposes and since the performance was very close, we average over all uplinks on a provider. Our results show that the best download times are possible via the direct peering link which directly connects the ISP to the OCH. While the download speed during the off periods is again close to the nominal speed of the client and does not vary across different ISPs, the download time improvements can be up to a factor of four during peak hours. This together with the observation of static uplink assignment, even during peak hours, shows that there is significant potential for PaDIS to improve end-user experience and enable the collaboration between ISPs and OCHs.

4.3 Video Streaming Providers

Video streaming via HTTP is popular and accounts for more than 20 % of the HTTP traffic. The leading video streaming provider in our trace, VSP1, is responsible for roughly 13 % of the overall HTTP traffic. We identify the architecture of VSP1 by examining our traces to analyze their naming scheme and perform a series of active measurements. We found that VSP1, in contrast to other CDS's, does not employ a DNS based load balancing scheme, but uses application layer redirection, namely HTTP 3xx return codes [16]. Additionally, we found that VSP1 uses a naming scheme that implies a caching hierarchy that organizes the caches of different layers into groups. While we have verified that all servers reply to any valid request, the reply might just be a redirection into the upper caching layer when the content is not available or even in a lower layer if the content is already well distributed. Another interesting observation we made is that VSP1 throttles the speed at which a user can download a video. After a short, but fast burst of data, the bitrate of the connection is throttled to roughly the video-bitrate. This behavior leads to virtually no performance difference of the caches as long as they can retain the video's bitrate. In this case, the PaDIS is not able to improve the perceived performance of an end-user.

Even though we found many unique cache IP addresses in our active measurements, they all belong to prefixes within the VSP1 AS and there seems to be limited path diversity for this provider. Still, we do believe this might be an interesting setting for PaDIS. On the one hand, a single path towards VSP1 is actually not as crucial as the the number of paths from VSP1 to the ISP, since the bulk of the transfered volume is flowing from VSP1 to the ISP. If the number of ingress paths exceeds the number of egress paths, PaDIS can be utilized to recommend caches such that the content is injected from the most appropriate ingress point in the ISP. On the other hand, in a collaborative scheme, VSP1 can utilize PaDIS to redirect clients to its caches taking into consideration network performance characteristics and replicate content in a more efficient manner.

4.4 Summary

We find that PaDIS has potential to substantially improve the end-user experience within an ISP by exploring the existing server and path diversity. We were able to show using active measurements in a big European ISP that PaDIS can significantly improve content delivery of some of the dominant content delivery platforms, including the two major CDNs and the top OCH, that are responsible for roughly 35 % of the overall HTTP traffic.

5. RELATED WORK

Content delivery networks are used to provide fast and scalable commercial-grade Web applications [23]. CDNs rely on large-scale commodity infrastructure to improve application performance. Krishnamurthy et al. [20] and later Huang et al. [17] characterize their performance, i. e., by quantifying the end-user performance, analyzing the DNS redirection overhead, unveiling the CDN server location, and assessing their availability.

Su et al. [34] propose to utilize CDN's redirection to locate high performance paths. Choffnes et al. [11] propose to use information from CDNs to bias neighbor selection in P2P systems without any path monitoring or probing.

Recently, Triukose et al. [35] show that most popular commercial CDNs as well as several community CDNs serve any object from any cache. They then use this insight to show that it is feasible to use the CDN's infrastructure to amplify attacks against CDN costumer web sites. Our work leverages this observation by including ISP information for cache site selection and thus improves end-user performance and enables ISP traffic engineering.

The ideas presented in this paper build upon our previous work on biasing peer selection in P2P systems [5, 4]. Our work also utilizes the insights from previous work [9] which has shown that server selection is important for enhancing the end-user experience.

To the best of our knowledge, this is the first work that proposes and deploys a system for ISP and application collaboration. It is based on the insight that today content is usually accessible from multiple locations. In the past, game-theoretic studies [18, 14] have investigated the principle possibilities of cooperation between ISPs and CDNs as well as the potential of an ISP deploying its own CDN. However, they have not proposed a system that enables this.

6. CONCLUSION

Our study, based on traces from more than 20,000 residential users as well as active DNS measurements, shows that there is significant server location diversity as well as path diversity for accessing HTTP based content. The key insight is that today most content delivery architectures rely on distributed infrastructures. We therefore propose and deploy PaDIS, a novel system that allows ISPs to discover and utilize path diversity. Using extensive active measurements from vantage points within a residential network, we were able to show the benefits that PaDIS can offer to the end-user experience. More specifically, we can show significant improvements in download times of up to a factor of four for content offered by the most popular content providers, including CDNs and OCHs, for users of an ISP. Our results also highlight the benefits for ISPs.

PaDIS may act as a catalyst for ISPs to regain control of their own traffic. PaDIS is a tool that can assist ISPs in performing traffic engineering or in driving up utilization for monetary gain at the application layer by biasing server selection for their customers. Furthermore, it can also serve as a negotiation tool between ISPs or between ISPs and content providers. In addition, it might be used not only as a means of cooperation but also for revenue sharing. Content providers and distributors may also utilize PaDIS to enhance content replication and delivery strategy due to increased access to meta-information.

7. ACKNOWLEDGMENTS

This work was supported in part by a grant from Deutsche Telekom Laboratories and the BMBF project G-Lab. We would also like to thank our shepherd, Robert Beverly, and the anonymous reviewers for their constructive comments.

8. REFERENCES

[1] Hrishikesh B. Acharya and Mohamed G. Gouda. The Theory of Network Tracing. In *Proc. of ACM PODC '09*.

[2] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing DNS Resolvers in the Wild. In *Proc. of ACM IMC '10*.

[3] Bernhard Ager, Fabian Schneider, Juhoon Kim, and Anja Feldmann. Revisiting cacheability in times of user generated content. In *Proc. of IEEE Global Internet Symp. '10*.

[4] Vinay Aggarwal, Obi Akonjang, and Anja Feldmann. Improving User and ISP Experience through ISP-aided P2P Locality. In *Proc. of IEEE Global Internet Symp. '08*.

[5] Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. Can ISPs and P2P Users Cooperate for Improved Performance? *SIGCOMM Comput. Commun. Rev.*, 37(3), 2007.

[6] Hussein A. Alzoubi, Michael Rabinovich, and Oliver Spatscheck. MyXDNS: A Request Routing DNS Server with Decoupled Server Selection. In *Proc. of WWW '07*.

[7] Demetres Antoniades, Evangelos P. Markatos, and Constantine Dovrolis. One-click Hosting Services: a File-sharing Hideout. In *Proc. of ACM IMC '09*.

[8] Ruchir Bindal, Pei Cao, William Chan, Jan Medved, George Suwala, Tony Bates, and Amy Zhang. Improving Traffic Locality in BitTorrent via Biased Neighbor Selection. In *Proc. of IEEE ICDCS '06*.

[9] Robert L. Carter and Mark E. Crovella. On the Network Impact of Dynamic Server Selection. *Computer Networks*, 31((23-24)):2529–2558, 1999.

[10] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. Analyzing the Video Popularity Characteristics of Large-scale User Generated Content Systems. *IEEE/ACM Trans. Networking*, 17(5):1357–1370, 2009.

[11] David R. Choffnes and Fabián E. Bustamante. Taming the Torrent: a Practical Approach to Reducing Cross-ISP Traffic in Peer-to-peer Systems. In *Proc. of ACM SIGCOMM '08*.

[12] C. Contavalli, W. van der Gaast, S. Leach, and D. Rodden. Client IP Information in DNS Requests. IETF draft, work in progress, draft-vandergaast-edns-client-ip-00.txt, Jan 2010.

[13] Graham Cormode and Marios Hadjieleftheriou. Methods for Finding Frequent Items in Data Streams. *The VLDB J.*, 19(1):3–20, 2010.

[14] Dominic DiPalantino and Ramesh Johari. Traffic Engineering versus Content Distribution: A Game-theoretic Perspective. In *Proc. of IEEE INFOCOM '09*.

[15] Holger Dreger, Anja Feldmann, Michael Mai, Vern Paxson, and Robin Sommer. Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In *Proc. of USENIX Security Symp. '06*.

[16] Roy Fielding, Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, Jun 1999.

[17] Cheng Huang, Angela Wang, Jin Li, and Keith W. Ross. Measuring and Evaluating Large-scale CDNs. In *Proc. of ACM SIGCOMM IMC '08, paper withdrawn*.

[18] Wenjie Jiang, Rui Zhang-Shen, Jennifer Rexford, and Mung Chiang. Cooperative Content Distribution and Traffic Engineering in an ISP Network. In *Proc. of ACM SIGMETRICS '09*.

[19] Thomas Karagiannis, Pablo Rodriguez, and Konstantina Papagiannaki. Should ISPs fear Peer-Assisted Content Distribution? In *Proc. of ACM IMC '05*.

[20] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the Use and Performance of Content Distribution Networks. In *Proc. of ACM SIGCOMM IMW '01*.

[21] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving Beyond End-to-end Path Information to Optimize CDN Performance. In *Proc. of ACM SIGCOMM IMC '09*.

[22] Craig Labovitz, Scott Iekel Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. Internet Inter-Domain Traffic. In *Proc. of ACM SIGCOMM '10*.

[23] Tom Leighton. Improving Performance on the Internet. *Commun. ACM*, 52(2):44–51, 2009.

[24] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *Proc. of ACM SIGCOMM IMC '09*.

[25] P. Mockapetris. Domain Names - Implementation and Specification. RFC 1035, Nov 1987.

[26] Jianping Pan, Y. Thomas Hou, and Bo Li. An Overview of DNS-based Server Selections in Content Distribution Networks. *Comput. Netw.*, 43(6):695–711, 2003.

[27] Vern Paxson. Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23–24), 1999.

[28] Sylvia Ratnasamy, Mark Handley, Richard Karp, and Scott Shenker. Topologically Aware Overlay Construction and Server Selection. In *Proc. of IEEE INFOCOM '02*.

[29] Sandvine Inc. 2009 global broadband phenomena. `http://www.sandvine.com/news/global_broadband_trends.asp`.

[30] Fabian Schneider. *Analysis of New Trends in the Web from a Network Perspective*. PhD thesis, Technische Universität Berlin, Mar 2010.

[31] Hendrik Schulze and Klaus Mochalski. Internet study 2008-9. `http://www.ipoque.com/resources/internet-studies`.

[32] Jan Seedorf and Eric W. Burger. Application-Layer Traffic Optimization (ALTO) Problem Statement. RFC 5693, Oct 2009.

[33] S. S. Siwpersad, Bamba Gueye, and Steve Uhlig. Assessing the Geographic Resolution of Exhaustive Tabulation for Geolocating Internet Hosts. In *Proc. of PAM '08*.

[34] Ao-Jan Su, David R. Choffnes, Aleksandar Kuzmanovic, and Fabián E. Bustamante. Drafting behind Akamai: Inferring Network Conditions based on CDN Redirections. *IEEE/ACM Trans. Netw.*, 17(6):1752–1765, 2009.

[35] Sipat Triukose, Zakaria Al-Qudah, and Michael Rabinovich. Content Delivery Networks: Protection or Threat? In *Proc. of ESORICS '09*.

[36] Haiyong Xie, Y. Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. P4P: Provider Portal for applications. In *Proc. of ACM SIGCOMM '08*.