ISSN 2229-5518

Usage of Incremental Provider-aided Distance Information System to Improve User Assignment from a Performance Perspective for Content-Delivery Networks (CDNs)

C.Kishor Kumar Reddy,
Dept. of C.S.E, VCE,
Samshabad, R.R (Dist), A.P, India.
Kishoar4u@yahoo.com,
Ph: +91 9493024236.

M.Venkateswarlu,
Asso.Professor (Dept. of CSE),
VCE, Samshabad, A.P, India.
venkateswarlu08@gmail.com,
Ph: +91-9505733789.

P.R Anisha
Dept. of C.S.E, VCE,
Samshabad, R.R (Dist), A.P, India.
anishanaidu.pushpala@gmail.com,
Ph: +91 7386808082

Abstract**—** This paper looks at Content-delivery networks (CDNs) originate a large fraction of Internet traffic; yet, due to how CDNs often perform traffic optimization, users aren't always assigned to the best servers for end-user performance and, in particular, suggests Provider-aided Distance Information System (PaDIS), which is a mechanism to rank client-host pairs based upon information such as RTT, bandwidth or number of hops. Headline figure, 70% of http traffic from a major European ISP can be accessed via multiple different locations. “Hyper giants” are defined as the large content providers such as google, yahoo and CDN providers which effectively build their own network and have content in multiple places. Quote: “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” also “there is choice between at least two subnets over 40% of the time”. To improve user assignment of CDNs, we propose and deploy the Provider-aided Distance Information System (PaDIS), which lets ISPs augment the CDN server by utilizing their unique knowledge about network conditions and user locations. Field tests show that using PaDIS can result in significant improvements in download time. Our field test results show that significant improvements in download time up to a factor of four for content offered by popular CDNs can be achieved when utilizing PaDIS. This paper discusses on the usage of PADIS architecture.

Keywords- Usage of PADIS, Internet traffic, Traffic optimization, ISP,HTTP,clent/server, *local and wide-area networks, Internet network architecture, server selection, DNS.*

I. INTRODUCTION

With the proliferation of the Internet, popular Web services often suffer congestion and bottlenecks due to large demands made on their services. Such a scenario may cause unmanageable levels of traffic flow, resulting in many requests being lost. Replicating the same content or services over several mirrored Web servers strategically placed at various locations is a method commonly used by service providers to improve performance and scalability. The user is redirected to the nearest server and this approach helps to reduce network impact on the response time of the user requests.

Content Delivery Networks (CDNs) provide services that improve network performance by maximizing bandwidth, improving accessibility and maintaining correctness through content replication. They offer fast and reliable applications and services by distributing content to cache or edge servers located close to users. A CDN has some combination of content-delivery, request-routing, distribution and accounting infrastructure.

II. CONTEXT DELIVERY NETWORKS

Content Delivery Networks (CDN), evolved first in 1998, replicate contents over several mirrored web servers (i.e., surrogate servers) strategically placed at various locations in order to deal with the *flash crowds*. Geographically distributing the web servers’ facilities is a method commonly used by service providers to improve performance and scalability. A CDN has some combination of a content-delivery infrastructure, a request-routing infrastructure, a distribution infrastructure and an accounting infrastructure. CDNs improve network performance by maximizing bandwidth, improving accessibility and maintaining correctness through content replication and thus offer fast and reliable applications and services by distributing content to cache servers located close to us

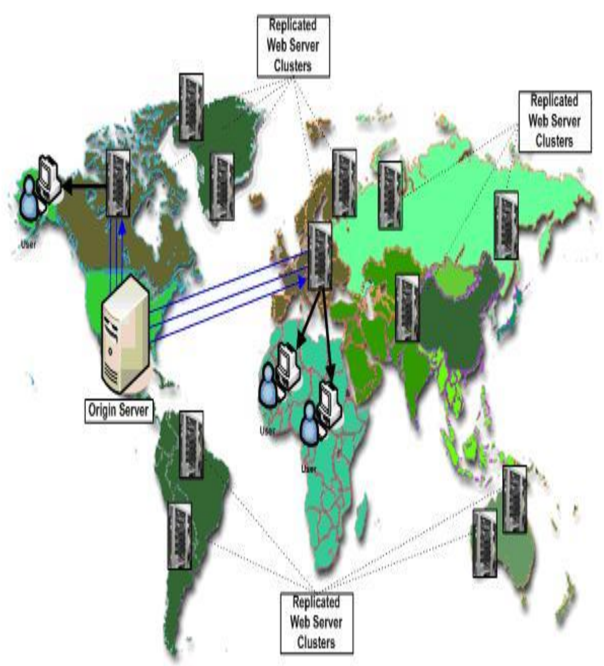


Figure 1: Abstract architecture of a Content Delivery Network (CDN)

Figure 1 shows a typical content delivery environment where the replicated Web server clusters are located at the edge of the network to which the end-users are connected. In such CDN environment, Web content based on user requests are fetched from the origin server and a user is served with the content from the nearby replicated web server. Thus the user ends up communicating with a replicated CDN server close to it and retrieves files from that server

Content/services provided by a CDN

CDN providers host third-party contents for fast delivery of any digital content, including static contents (e.g. Static HTML pages, images, documents, software patches etc), streaming media (e.g. audio, real time video etc) and varying content services (e.g. directory service, e-commerce service, file transfer service etc.). The sources of contents are large enterprises, web service providers, media companies, news broadcasters etc. The clients interact with the CDN specifying the content/service request through cell phone, smart phone/PDA, laptop, desktop etc. Figure 2 depicts the different content/services served by the CDN to different clients.

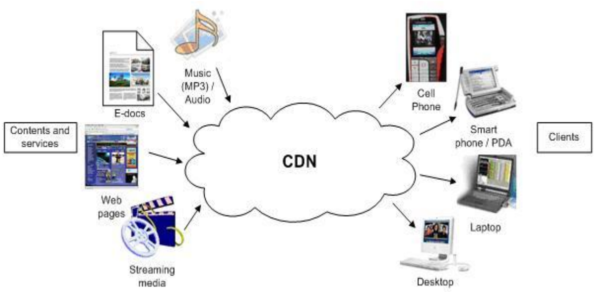


Figure 2: Content/services provided by a CDN

C. Basic interaction with CDN

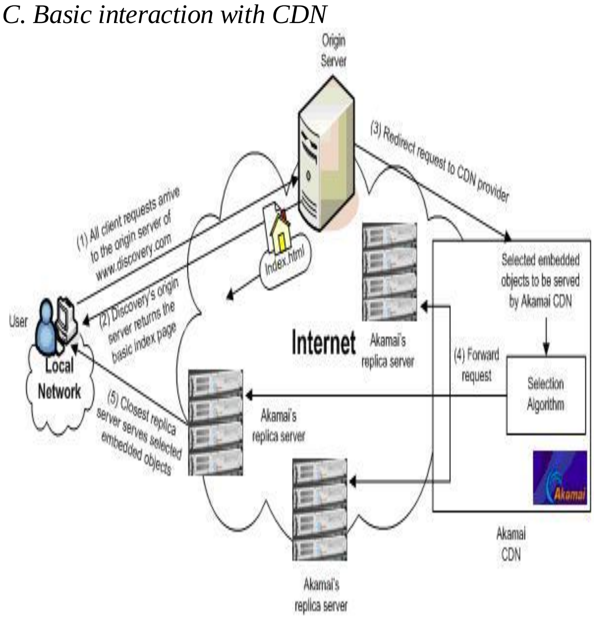


Figure 3: Basic interaction flows in a CDN environment

Figure 3 provides the high level view of the basic interaction flows among the components in a Content Delivery Network (CDN) environment. Here, discovery.com is the content provider and Akamai is the CDN that hosts the content of discovery.com. The interaction flows are: 1) the client requests content from www.discovery.com by specifying its URL in the Web browser. Client's request is directed to the origin server of discovery.com; 2) when discovery.com receives a request, its web server makes a decision to provide only the basic contents (e.g. index page of the site) that can be served from its origin server; 3) to serve the high bandwidth demanding and frequently asked contents (e.g. embedded objects – fresh content, navigation bar, banner ads etc. Figure 4 shows such a web page which contains the embedded objects served by Akamai CDN), discovery's origin server redirects client's request to the CDN provider (Akamai, in this case); 4) using the proprietary selection algorithm, the CDN provider selects the replica server which is 'closest' to the client, in order to serve the requested embedded objects; 5) selected replica server gets the embedded objects from the origin server, serves the client requests and caches it for subsequent request servicing.

Figure 4: Typical embedded web page contents served by Akamai CDN

III. How to configure wordpress to use a cdn

Research shows that if your web pages take longer than 5 seconds to load, you lose 50% of your viewers and sales. You can speed up your wordpress blog by using a CDN to display content to users faster and more efficiently. You can distributes common files or content such as css, javascript, uploaded images, videos and much more through a CDN, which serves the content from the closest cdn edge server to the end-user. In this tutorial, We will explains how to configure WordPress, Apache/Lighttpd webserver, Bind dns server to use a CDN to distribute your common files such as css, js, user uploaded files and lighten load on your web server.

Technology

CDNs may be implemented using commercially available software, with Aflexi, Inc. and Cisco Systems being prominent vendors, but larger CDNs usually run on customized platforms. Most web platforms include tools for integration with a CDN, such as the W3 Total Cache plug in for the Word Press platform.[4]

CDN nodes are usually deployed in multiple locations, often over multiple backbones.

A General Architecture of a DNS-based Server

A DNS-based server selection mechanism assigns end-users to servers whenever a request for content hosted by the CDN is submitted. To retrieve the content from the CDN, the end-user first resolves the hostname associated with the content. Once the hostname is resolved to an IP address, a HTTP connection to the supplied IP address is opened and the content is retrieved.

The general architecture of the DNS-based approach is presented in Figure 1. From the viewpoint of the end-users and ISPs, the redirection schemes employed by existing CDNs have three major limitations:

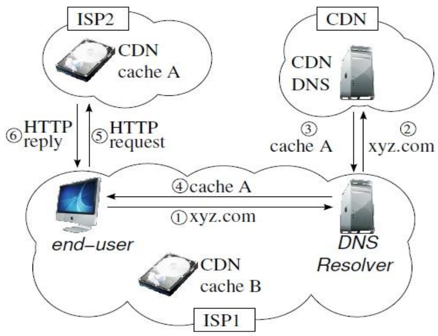


Figure 1.A General Architecture of A DNS-based Server

Network Bottlenecks:

Despite the traffic flow optimization performed by CDNs, the assignment of end-user requests to servers by CDNs may still result in sub-optimal content delivery performance for the end-users. This is a consequence of the limited information CDNs have about the network conditions between the end-user and their servers. Tracking the ever changing conditions in networks, i.e., through active measurements and end-user reports, incurs an overhead for the CDN without a guarantee of performance improvements for the end-user. Without sufficient information about the network paths between the CDN servers and the end-user, any assignment performed by the CDN may lead to additional load on existing network bottlenecks, or to the creation of new bottlenecks.

User Mislocation:

DNS requests received by the CDN DNS servers originate from the DNS resolver of the end-user, not from the end-user itself. The assignment is therefore based on the assumption that end-users are close to their DNS resolvers. Recent studies have shown that in many cases this assumption does not hold. As a result, the end-user is mis-located and the server assignment is not optimal. As a response to this issue, DNS extensions have been proposed to include the end-user IP information.

Content Delivery Cost:

Finally, CDNs strive to minimize the overall cost of delivering huge amounts of content to end-users. To that end, their assignment strategy is mainly driven by economic aspects. While a CDN will always try to assign users in such a way that the server can deliver reasonable performance, this can again result in end-users not being directed to the server able to deliver best performance. To overcome the limitations of current CDNs, we design and evaluate a Provider-aided Distance Information System (PaDIS). PaDIS is a novel system operated by an ISP that allows ISPs to improve end-user experience by utilizing information about the network bottlenecks and end-user location. We believe that a system like PaDIS fills a gap in the content delivery landscape, especially by taking into account ISP constraints and end-user performance.

IV. OVERVIEW AND HISTORY OF PaDis

A. Overview

Today's content delivery landscape is mostly unaware of the information ISPs have about dynamic network conditions and end-user location in the network. Through a Provider aided Distance Information System (PaDIS), we propose to let the ISP influence the server selection by extending the ISP DNS infrastructure. To improve end-user performance, PaDIS leverages the server diversity of CDNs available through the multiple locations from which a given content can be obtained as well as network information only available to the ISP.

We show that there is significant server diversity in the currently deployed CDNs. We find that more than 70% of the HTTP traffic in a large European ISP can be obtained from at least three different network locations. We also find that only seven CDNs are responsible for more than 50% of the HTTP traffic. Moreover, large-scale studies show that popular CDNs allow end-users to download the content they host from any server. To leverage the available server diversity, PaDIS uses network information only available to the ISP to give recommendations on which server is best chosen from a network perspective. Our evaluation of PaDIS provides evidence that the knowledge of the ISP can improve the CDN server assignment process by avoiding the current limitations of DNS-based server selection.

B. History

Recent studies report that HTTP is used to access this information and it accounts for more than 50%of the traffic. Among the major causes for the current prevalence of HTTP traffic, we find the increase of streaming content, e.g., offered by youtube.com, as well as the popularity of the content offered by One- Click Hosters (OCHs) such as rapidshare.com. This popular content is hosted by the new "Hyper Giants" which include large content providers (CPs), such as Google and Yahoo!, as well as Content Distribution Networks (CDNs), such as Akamai and Limelight.

To achieve high levels of performance and scalability, CDNs rely on distributed infrastructures. Some of them even have deployed servers deep inside ISPs in more than 5000 locations throughout the Internet. Two techniques to direct end-users to servers are used by CDNs. These are DNS-based or work by HTTP redirection. Since HTTP redirection yields a higher overhead and is more intrusive into the architecture of the application than DNS-based schemes, the latter is more widely used.

V. MODELLLING AND IMPLEMENTATION

A. Deployment:

Deploying PaDIS inside an ISP is a two-step incremental process. First, the DNS infrastructure needs to be modified to be able to communicate with PaDIS. As ISPs often operate multiple DNS resolvers this has to be done step by step to keep interruptions to a minimum. Furthermore, leveraging the DNS process requires the end-users to utilize the ISP DNS resolver instead of a third party DNS resolver, e. g., Open DNS or Google DNS. To that end, we verify in a large European ISP that more than 97% of the customers use the DNS resolver supplied by the ISP.Second, PaDIS can be configured to influence the server selection of all CDNs or only the server selection of some of them, e.g. the most popular. Given that CDNs are responsible to deliver different types of content, e. g., real-time content, websites and/or bulk data, the administrator of PaDIS can set diverse performance metrics for different CDNs.

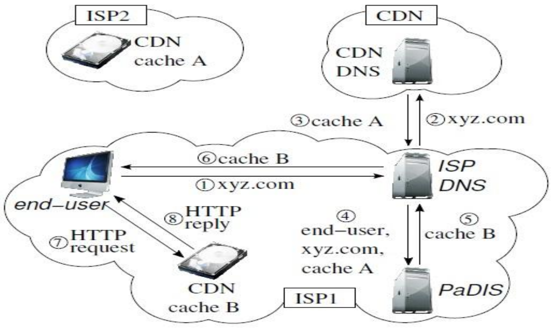


Figure 2: The incremental PaDIS deployment

The incremental PaDIS deployment adds two additional steps to the DNS resolution process: the DNS infrastructure of the ISP forwards the authoritative answer to the PaDIS server (4), which incorporates the answer into the server diversity, ranks all available servers based on the network information and then returns the best server to the DNS resolver of the end-host (5) which in turn forwards the answer to the end-host.

B. Architecture

To achieve the tasks of PaDIS we propose an architecture that comprises of a Diversity Discovery component, a Network Monitoring component and a Query Processing engine. In Figure 3 we provide the architecture overview of PaDIS.

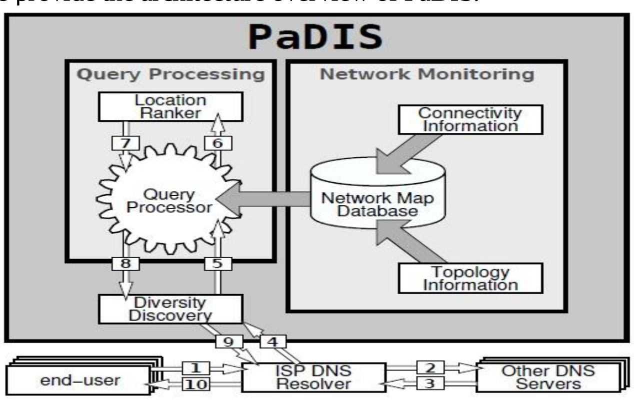


Figure3. PaDIS Architecture

(i) Diversity Discovery:

Improving the assignment of end-users to servers requires knowledge about the location diversity of content servers. However, CDNs do not provide a list of the the content servers they operate and their location. The Diversity Discovery component extracts CDN server diversity from DNS replies observed by the ISP DNS resolver and applies aggregation rules defined by the administrator of PaDIS. The Aggregation rules filter and build lists of IP addresses of servers able to satisfy a user request for content. With this, they enhance the choice offered to the query processing beyond the scope of the individual DNS reply. The Diversity Discovery component also acts as a protocol converter providing an interface between the ISP DNS resolver and the other components of PaDIS. It speaks the standard DNS protocol with the ISP DNS resolver, reducing the needed changes in ISP DNS resolver.

(ii) Network Monitoring:

ISPs have very detailed and up-to-date information about their own network infrastructure. Nonetheless, finding out the overall network status involves several systems each one responsible to monitor a part of the network status. The Network Monitoring component gathers information about the state of the network from several sources and maintains an up-to-date view of the network. It also provides an interface for network status queries. The Network Monitoring component consists of three sub-components:

(iii) Query Processing:

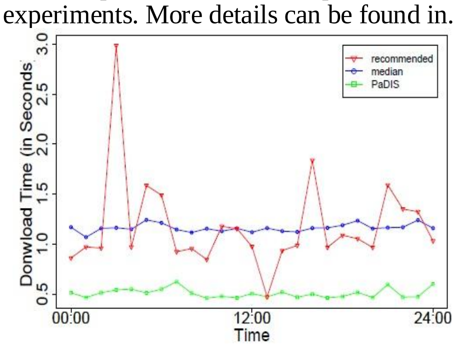
The Query Processing component combines the information about the server diversity as well as the up-to-date network status to improve content delivery. Its function is to calculate the preference of a given set of candidate sources (servers) to reach a destination (end-user) while taking into account the network conditions between the sources and the destination. It comprises of two sub-components. The Query Processor and the Location Ranker.

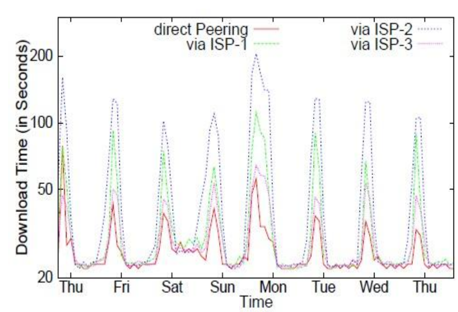
The Query Processor augments each source-destination pair in the query with network conditions. Then, it hands each pair individually to the Location Ranker to get its preference value. Finally, the list of source destinations pairs is sorted by its preference values and handed back to the Diversity Discovery.

The Location Ranker receives exactly one source-destination pair along with the network conditions between them. It applies a specific metric to be optimized, i.e., delay or link utilization, to calculate a preference value.

(iv) Performance Evaluation:

To quantify the effect of PaDIS, we consider two different CDNs. Our choice is driven by the large traffic volume they carry in the major European ISP we consider. With the help of anonymized packet level traces from the large European ISP, we select a large CDN, responsible for more than 20 % of the HTTP traffic in the studied ISP, with highly distributed caches that typically deliver small to average size files. The second is a OCH, responsible for more than 15% of the HTTP traffic in the studied ISP, relying on a multihomed data-center to deliver large size files to end-users. Once the CDNs were chosen, we deployed vantage points at residential locations with DSL connectivity in the large European ISP that operates PaDIS and performed extensive active measurements to evaluate PaDIS in the wild. The results presented here are representative of a much larger set of experiments. More details can be found in.





Top: Downloading a 510 Kbytes file distributed by a CDN: (recommended) download time when the cache recommendation from the CDN is used, (median) the download time of all available CDN caches as discovered by PaDIS, (PaDIS) download time when using PaDIS.

Bottom: Downloading a 50 Mbytes file from a OCH: PaDIS utilizes the direct peering with OCH improving the download time by a factor of four during the peak hour.

For the CDN, we utilize the Diversity Discovery component of PaDIS to extract the server diversity. The aggregated rule we are using is based on the DNS redirection signature of the CDN. We were able to identify more than 3500 unique servers operated by the large CDN spread in over 124 different network locations. A large fraction of these caches are located within the studied ISP. Next, we randomly selected one cache from each location in order to run our experiments and confirmed that all the chosen caches all serve the content we are downloading for our measurement. Note that most CDNs server all content from all their caches. As most of the files delivered by this CDN have small to average size, the download performance is dominated by the end-to-end delay. Thus, we use end-to-end delay as the metric to rank caches in PaDIS. During the experiment, we downloaded several files from all 124 chosen servers, as well as from the server chosen by the CDN and the server chosen by PaDIS. Figure 4 (top) shows the download time of an average size object. We also plot the median download time across all the 124 pre-selected caches. The reduction in download time was up to a factor of four when using PaDIS. When considering larger files, e. g., more than 10 Mbytes, the improvement in download time is smaller as the download performance is restricted by the network bandwidth on the bottleneck link and the end-to-end delay becomes less significant.

We now turn our attention to the OCH. Using the same approach as with the CDN, we were able to uncover that the OCH servers are located in a single data-center. The OCH is well connected to four large ISPs, including the large European ISP that operates PaDIS. A closer look at the server selection strategy of the OCH reveals that it assigns 60% of the requests originated by the end-users of the studied ISP to servers behind the direct peering with it regardless of the time of day or the day of week. The remaining requests are randomly assigned to the servers behind the peerings with the other three ISPs. Knowing this, we set the ranking function of PaDIS such that OCH servers that are behind the direct peering with the large European ISP are ranked higher than the others. This way, PaDIS always prefers the direct peering. We performed an experiment where we repeatedly downloaded content served using the different peerings of the OCH. In Figure 4 (bottom) we plot the download time of a 50 Mbytes file during the period of one week. The improvement of the download time was up to a factor of four during the peak hour when the network resources are limited.

(v) PaDis Usage options

While PaDIS can be used as an ALTO 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 bythe round-trip-time or bandwidth, e. g., for One-Click Hosters 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

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 toits 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 Cross Query 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. 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.

VI. 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 ISPsTo improve end-user performance; we propose a solution where the ISP operates a Provider-aided Distance Informa- tion System (PaDIS). PaDIS uses information that is only available to the ISP, in order to rank any user-server pair based on network conditions and user location. As a result, PaDIS utilizes the deployed CDNs infrastructure to improve end-user performance. We evaluate PaDIS with different providers and show that improvements in download times of up to a factor of four are possible.

The expected gains from deploying PaDIS depend on several aspects, such as the number of locations from which a given content is available, the properties of the paths avail- able between the servers and the end-users, as well as the type and volume of the requested content. CDNs continue to expand their infrastructures to keep up with the increas ing demand of traffic. Thus, the gain of deploying PaDIS is expected to grow in the future as it can take advantage of the increased server diversity exposed by the CDNs.

PaDIS leverages the decoupling of the server selection and the content transfer, thus it is not restricted to DNS-based server selection and "CDNized" traffic, but can be utilized by any protocol. As part of our future work we would like to investigate how PaDIS can be utilized to improve the content delivery performance of applications that do not depend on DNS as well as evaluating PaDIS in other ISPs.

References:

1. Erik Nygren, Ramesh K. Sitaraman, and Jennifer Sun. "The Akamai Network: A Platform for High-Performance Internet Applications, ACM SIGOPS Operating Systems Review, Vol. 44, No.3, July 2010." .
2. "Akamai goes P2P, buys Red Swoosh. GigaOm" . April 2, 2007. Retrieved March 16, 2012.
3. http://www.windowsazure.com/enus/home/features/cdn/
4. http://wordpress.org/extend/plugins/w3-totalcache/
5. Saltzer, J. H., Reed, D. P., Clark, D. D.: “End-to-End Arguments in System Design,” ACM Transactions on Communications, 2(4), 1984
6. Hofmann, Markus; Leland R. Beaumont (2005). *Content Networking: Architecture, Protocols, and Practice*. Morgan Kaufmann Publisher. ISBN 1-55860-834-6.
7. RFC 3568 Barbir, A., Cain, B., Nair, R., Spatscheck, O.: "Known Content Network (CN) Request-Routing Mechanisms," July 2003
8. RFC 1546 Partridge, C., Mendez, T., Milliken, W.: "Host Anycasting Services," November 1993.
9. RFC 3507 Elson, J., Cerpa, A.: "Internet Content Adaptation Protocol (ICAP)," April 2003.
10. Application-Layer Traffic Optimization (ALTO) IETF working group. https://datatracker.ietf. Org/wg/alto/charter/.