Enabling Content-aware Traffic Engineering

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

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

Steve Uhlig
Queen Mary, U. London
steve@qmul.eecs.ac.uk

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

Bruce Maggs
Duke/Akamai
bmm@cs.duke.edu

Abstract

Today, a large fraction of Internet traffic is originated by Content Delivery Networks (CDNs). To cope with increasing demand for content, CDNs have deployed massively distributed infrastructures. These deployments pose challenges for CDNs as they have to dynamically map end-users to appropriate servers without being fully aware of the network conditions within an Internet Service Provider (ISP) or the end-user location. On the other hand, ISPs struggle to cope with rapid traffic shifts caused by the dynamic server selection policies of the CDNs.

The challenges that CDNs and ISPs face separately can be turned into an opportunity for collaboration. We argue that it is sufficient for CDNs and ISPs to coordinate only in server selection, not routing, in order to perform traffic engineering. To this end, we propose Content-aware Traffic Engineering (CaTE), which dynamically adapts server selection for content hosted by CDNs using ISP recommendations on small time scales. CaTE relies on the observation that by selecting an appropriate server among those available to deliver the content, the path of the traffic in the network can be influenced in a desired way. We present the design and implementation of a prototype to realize CaTE, and show how CDNs and ISPs can jointly take advantage of the already deployed distributed hosting infrastructures and path diversity, as well as the ISP detailed view of the network status without revealing sensitive operational information. By relying on tier-1 ISP traces, we show that CaTE allows CDNs to enhance the end-user experience while enabling an ISP to achieve several traffic engineering goals.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.3 [**Network Operations**]: Network Management

Keywords

Traffic Engineering, Content Distribution, Network Optimization.

1. INTRODUCTION

Recent traffic studies [14, 18, 23] show that a large fraction of Internet traffic is originated by a small number of Content Delivery Networks (CDNs). Major CDNs are popular media sites like YouTube and Netflix, One-Click Hosters (OCHs), e.g., RapidShare, commercial CDNs such as Akamai and Limelight, and content providers, e.g., Google, Yahoo!, and Microsoft. Gerber and Doverspike [14] report that a few CDNs account for more than half the traffic of a US-based tier-1 carrier. Poese et al. [23] report a similar observation from the traffic of a European tier-1 carrier. Labovitz et al. [18] infer that more than 10% of the total Internet inter-domain traffic originates from Google, and Akamai claims to deliver more than 20% of the total Internet Web traffic [21]. In North America, Netflix is responsible for around 30% of the traffic during peak hours [15] and utilizes multiple CDNs.

To cope with the increasing demand for content, CDNs deploy massively distributed server infrastructures [19] to replicate content and make it accessible from different locations in the Internet [29, 3]. For example, Akamai operates more than 80 000 servers in more than 1 800 locations across nearly 1 000 networks [19, 21]. Google is reported to operate tens of data-centers and front-end server clusters worldwide [17, 28]. Microsoft has deployed its CDN infrastructure in 24 locations and Amazon maintains at least 5 large data-centers and caches in at least 21 locations. Limelight operates thousands of servers in more than 22 delivery centers and connects directly to more than 900 networks worldwide.

The growth of demand for content and the resulting deployment of content delivery infrastructures pose new challenges to CDNs and to Internet Service Providers (ISPs). For CDNs, the cost of deploying and maintaining such a massive infrastructure has significantly increased during the last few years [25] and the price charged for delivering traffic to end-users has decreased due to the intense competition. CDNs also struggle to engineer and manage their infrastructures, replicate content based on end-user demand, and assign end-users to appropriate servers. The latter is challenging due to the mis-location of end-users [20, 2]. Furthermore, inferring the network conditions within an ISP without direct information from the network is difficult [21]. Moreover, due to highly distributed server deployment and adaptive end-user to server assignment, the traffic injected by CDNs is volatile. For example, if one of its server locations is overloaded, a CDN will re-assign end-users to other server locations, resulting in large traffic shifts in the ISP network within minutes.

We argue that the challenges that both CDNs and ISPs face separately can be turned into an opportunity if CDNs and ISPs are able to collaborate on the end-user to server assignemnt (1) on small time scales (minutes or even seconds), (2) by fully utilizing the CDN server and network path diversity that is currently available, and (3) with minimal overhead in the current operation of CDNs and ISPs. Today, no traffic engineering system supports all the above three requirements.

Routing-based traffic engineering adjusts routing weights to adapt to traffic matrix changes. To avoid micro-loops during routing convergence, it is common practice to only adjust a small number of routing weights [11]. To limit the number of changes in routing weights, routing-based traffic engineering relies on traffic matrices computed over long time periods and offline estimation of the routing weights. Therefore, routing-based traffic engineering operates on time scales of hours, which can be too slow to react to rapid change of traffic demands. Only recently, link-weight tweak-

ing [12] approaches were introduced to safely adapt weights on smaller time scales.

Other traffic engineering proposals [27, 9, 10] split traffic over multiple paths, and thus, operate on small time scales (minutes). These proposals require changes to the routers or use of proprietary protocols supported by router ventors. It may also require the maintenance of state of individual TCP connections. Two of the proposed solutions [9, 27] are restricted to MPLS-like solutions.

All the above approached are ISP-sided, meaning that the ISP utilizes them to react on the decision made by the CDN. Recently, CDN-ISP collaboration approaches [8, 16, 30, 4] were introduced to perform routing-based traffic engineering. Portals that have been proposed for peer-to-peer applications and users to communicate with network providers and get an updated view of their networks [31] can also be applied. In some proposals [4, 31] peer-to-peer content is taken into consideration for traffic engineering decisions, yet ISPs have to frequently reveal updated information about the topology and the operation of their networks. This requires large amount of sensitive information to be exchanged between two parties and hands control over traffic engineering to the client side, which makes it hard to be adopted by ISPs. In other cases [16] the solution is limited to the cooperation between the ISP and the CDN that is operated by same ISP.

In this paper we argue that CDN and ISP collaboration only in server selection, not routing, is sufficient for traffic engineering and end-user experience improvement. This is a clear differentiator from previous works on CDN-ISP collaboration [8, 16, 26, 30, 4, 31] that focus on routing. To this end, we introduce and evaluate a novel traffic engineering method, that we call *Content-aware Traffic Engineering* (CaTE) designed to respect all the above three requirements (operation in small time scales, full utilization of server and path diversity and minimal operational overhead). Our contributions can be summarized as follows:

- We introduce the concept of CaTE: Instead of relying on proposed traffic engineering techniques that require changes in routing or multi-path, we argue that it is sufficient to explore the existing server diversity. By collaborative CDN-ISP server selection it is possible to re-direct end-users to servers on small time scales and change the traffic matrix as desired. We introduce CaTE in Section 2.
- We design and implement a CaTE prototype system that can scale up to thousands of requests per second (per instance of the system) thanks to its novel design on how to gather and maintain network data, as well as the interaction between the proposed ISP and existing CDN query processing systems. The performance of this system is far beyond the capabilities of the current state-of-the-art [4] and does not add overhead in the operation of CDNs and ISPs (see Section 3).
- We provide the first of its kind evaluation of a CaTE system with operational data from a large tier-1 ISP and a number of CDNs under different performance metrics. We report the benefits for CDNs, ISPs and end-users in Section 4.

2. THE CaTE APPROACH

The increasing challenges of cost reduction and end-user experience improvement that both CDNs and ISPs are confronted with, motivate us to propose a new tool in the traffic engineering landscape. We introduce *Content-aware Traffic Engineering* (CaTE). CaTE leverages the server location diversity offered by CDNs and, through this, enables adaptation to traffic demand shifts. In fact, CaTE relies on the observation that by selecting an appropriate server among those available to deliver the content, the path of the traffic in the network can be influenced in a desired way. Recent measurement studies [23] show that there is significant server and path diversity that is at large unexplored. Figure 1 illustrates the basic concept of CaTE. The content requested by the client is in principle available from three CDN servers (A, B, and C). However, the client only connects to one of the network locations. Today, the decision of where the client connects to is solely done by the CDN and is partially based on measurements and/or inference of network information and end-user location. With CaTE the decision on end-user to server assignment can be enhanced by utilizing ISP recommendations.

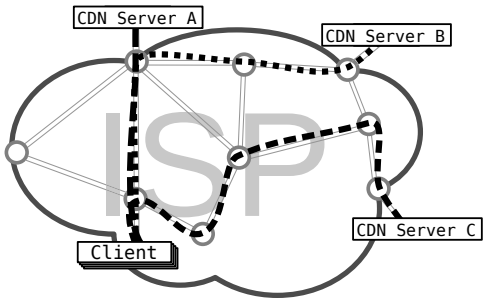

Figure 1: By choosing a CDN server for a client with the help of CaTE, traffic engineering goals and accurate end-user to server assignment become possible.

CaTE complements existing traffic engineering solutions [11, 12, 27, 9, 10, 8, 16, 26, 30, 4, 31, 27, 9, 10] by focusing on server selection rather than routing. Let $\mathbf{y}$ be the vector of traffic counts on links and $\mathbf{x}$ the vector of traffic counts in origin-destination (OD) flows in the ISP network. Then $\mathbf{y}=A\mathbf{x}$, where A is the routing matrix. $A_{ij} = 1$ if the OD flow i traverses link j and 0 otherwise. Traffic engineering is the process of adjusting A, given the OD flows $\mathbf{x}$, so as to influence the link traffic $\mathbf{y}$ in a desirable way. In CaTE, we revisit traffic engineering by focusing on traffic demands rather than routing changes:

Definition 1: Content-aware Traffic Engineering (CaTE) is the process of adjusting the traffic demand vector $\mathbf{x}$, given a routing matrix A, so as to change the link traffic $\mathbf{y}$.

Only traffic for which server location diversity exists can be adjusted. Therefore, $\mathbf{x}=\mathbf{x}_r+\mathbf{x}_s$ where $\mathbf{x}_r$ and $\mathbf{x}_s$ denote the content demands that can and can not be adjusted (single server location case) respectively. The degree of freedom in adjusting traffic highly depends on the diversity of locations from which the content can be obtained. We can rewrite the relation between traffic counts on links and traffic counts in flows as follows: $\mathbf{y}=A(\mathbf{x}_s + \mathbf{x}_r)$. CaTE adjusts the traffic on each link of the network by adjusting the content demands $\mathbf{x}_r$: $\mathbf{y}_r=A\mathbf{x}_r$ to satisfy traffic engineering goals.

In CaTE the redirection of end-users to servers takes place on small time scales. Typical timescales for operation can be in the order of minutes, but in principle, it is possible to operate on even smaller time scales if applications or network conditions require, e.g., to react to flash crowds. As we will show in detail in Section 3.4 it is possible to operate on the time scales of the TTL value of a DNS query that is typically tens of seconds in large CDNs [23] or even per request. Notice that CaTE can be applied to CDNs that are operated by the ISP or to any other CDN.

Thanks to the online recommendations by ISP networks, CDNs gain the ability to better assign end-users to servers and better amortize the deployment and mantainance cost of their infrastructure. Network bottlenecks are also circumvented and thus the ISP operation is improved. Furthermore, the burden of measuring and inferring network topology, and the state of the network, both challenging problems, is removed from CDNs. Moreover, in [13, Sections 4 and 5] we show that the online CaTE decisions on the end-user to server assignment lead to optimal traffic assignment within the

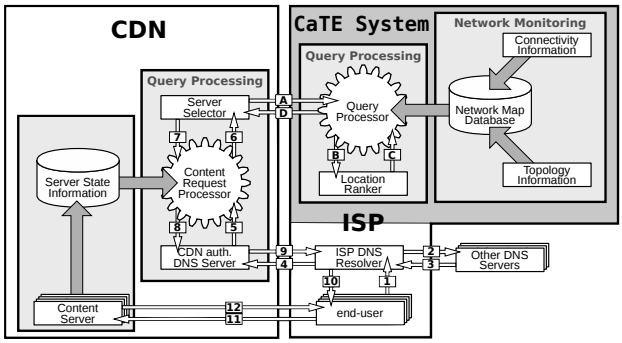

Figure 2: CaTE System architecture and flow of messages.

network under a number of different metrics. The advantage is that now the problem of assigning traffic to links reduces to a fractional solution (on the contrary, assigning routing weights to links is NP-hard). In short, all involved parties, including the end-users, benefit from CaTE, creating a win-win situation for everyone.

3. A PROTOTYPE TO SUPPORT CaTE

CaTE relies on a close collaboration between CDN and ISP on small time scales (minutes or even seconds). To achieve this goal, network information has to be collected and processed by the ISP. Candidate CDN servers have to be communicated to the ISP and ranked based on a commonly agreed criteria, e.g., to optimize the delay between the end-user and the CDN server. Today, there is no system to support the above operations. This motivates us to design, implement and evaluate a novel and scalable system that can support CaTE. In this section we describe the architecture and deployment of our working prototype to enable CaTE. We start by presenting our prototype in Section 3.1. We then comment on its operation and deployment within the ISP, its interaction with a CDN, and its performance that is beyond the state-of-the-art [4].

3.1 Architecture

The CaTE system is installed in an ISP and interacts with the existing CDN server selector. The main tasks of the CaTE system are to: (1) maintain an up-to-date annotated map of the ISP network and its properties, (2) produce preference rankings based on the paths between end-users and candidate servers, and (3) communicate with the CDN server selector to influence the assignment of end-user to servers. To this end, we propose an architecture that comprises a *Network Monitoring* component, a *Query Processing* component and a *Communication Interface* between an ISP and a CDN. For an overview of the architecture see Figure 2.

3.1.1 Network Monitoring

The network monitoring component gathers information about the topology and the state of the network from several sources to maintain an up-to-date view of the network. The network monitoring component consists of the following subcomponents:

The **Topology Information** component gathers detailed information about the basic network topology, i.e., routers and links, as well as annotations such as link utilization, router load, and topological changes. An Interior Gateway Protocol (IGP) listener provides up-to-date link-state (i.e., IS-IS, OSPF) information. Information about routers and links is retrieved, thus, the network topology can be extracted. The nominal link delay, i.e., the latency on a link without queuing, can be found through the link length and physical technology. The link utilization and other metrics can be retrieved via SNMP from the routers or an SNMP aggregator.

The **Connectivity Information** component uses routing information to calculate the paths that traffic takes through the network. Finding the path of egress traffic can be done by using a Border Gateway Protocol (BGP) listener. Ingress points of traffic into the ISP network can be found by utilizing Netflow data. This allows for complete forward and reverse path mapping inside the ISP. Furthermore, the system can map customers as well as CDN infrastructures into the network map by finding the routers that announce the address space associated with them. In total, this allows for a complete path map between any two points in the ISP network. Finally, our system has access to an uplink database that provides information about the connectivity statistics of end-users.

The **Network Map Database** component processes the information collected by the *Topology* and *Connectivity Information* components to build an annotated network map of the ISP network tailored towards fast lookup on path properties. It uses a layer of indirection to keep the more volatile information learned from BGP separate from the slower changing topological information. This allows address space to be quickly reassigned without any reprocessing of routing or path information. It also enables precalculation of path properties for all paths that yields a constant database lookup complexity independent of path length and network architecture. If topology changes, e.g., IGP weights change or a link fails, the *Topology Information* component immediately updates the database which only recalculates the properties of the affected paths. Having ISP-centric information ready for fast access in a database ensures timely responses and high query throughput.

3.1.2 Query Processing

The **Query Processing** component receives a description of a request for content from the CDN, which specifies the end-user making the request and a list of candidate CDN servers. It then uses information from the *Network Map Database* and a selected ranking function to rank the candidate servers. This component consists of the following subcomponents:

The **Query Processor** receives the query from the CDN. First, the query processor maps each source-destination (server to end-user) pair to a path in the network. In most cases, the end-user is seen as the ISP DNS resolver, unless both ISP and CDN support the client IP eDNS extension [7]. Once the path is found, the properties of the path are retrieved. Next, the pairs are run individually through the location ranker subcomponent (see below) to get a preference value. Finally, the list is sorted by preference values, the values are stripped from the list, and it is sent back to the CDN.

The **Location Ranker** component computes the preference value for individual source-destination pairs based on the source-destination path properties and an appropriate function. Which function to use depends on (a) CDN deployment and load, (b) CDN preferences and (c) the optimization goal of the ISP. The preference value for each source-destination pair is then handed back to the Query Processor. Multiple such optimization functions being defined upon the collaboration agreement between a CDN and an ISP, and subsequently selected individually in each ranking request. For example, a function might be the minimization of end-user and server delay. In Section 4 we evaluate CaTE with multiple ranking functions for different optimization goals.

3.1.3 Communication Interfaces

When a CDN receives a content request, the *Server Selector* needs to choose a content server to fulfill this request. We propose that the server selector sends the list of eligible content servers along with the source of the query and an optimization goal to the ISP's CaTE system to obtain additional guidance about the underlying network. If the guidance is at granularity of a single DNS

request, we propose a DNS-like protocol using UDP to prevent extra overhead for connection management. If the granularity is at a coarser level, i.e. seconds or even minutes, we rely on TCP.

3.2 Privacy and Performance

During the exchange of messages, none of the parties is revealing any sensitive operational information. CDNs only reveal the candidate servers that can respond to a given request without any additional operational information (e.g., CDN server load, cost of delivery). The set of candidate servers can be updated per request or within a TTL that is typically in the order of a tens of seconds in popular CDNs [23]. On the other side, the ISP does not reveal any operational information or the preference weights it uses for the ranking. In fact, the ISP only re-orders a list of candidate servers provided by the CDN. This approach differs significantly from [4, 31], where partial or complete ISP network information, routing weights, or ranking scores are publicly available. We argue that an important aspect to improve content delivery is to rely on up-to-date information during server selection of the CDN. This also eliminates the need of CDNs to perform active measurements to infer the conditions within the ISP that can add overhead to CDN operation and may be inaccurate. With CaTE, the final decision is still made by the CDN, yet it is augmented with up-to-date network guidance from the ISP.

To improve the performance of our system, we do not rely on XML-based network maps as proposed in [4], but on light protocols that are similar to DNS. This design choice is important as topology information in large networks is in the order of tens of MBytes. Transferring this information periodically to many end-users is likely to be challenging. In a single instance of our system, we manage to reply to up to 90 000 queries/sec when 50 candidate servers supplied by the CDN. At this level, the performance of our system is comparable to that of current DNS servers, such as BIND. However, the number of replies drops to around 15 000 per second when considering 350 candidate servers. The additional response time when our system is used is around 1 ms when the number of candidate servers is 50 and around 4 ms when considering 350 candidate servers. This overhead is small compared to the DNS resolution time [2]. The performance was achieved on a commodity dual-quad core server with 32 GB of RAM and 1GBit Ethernet interfaces. Furthermore, running additional servers does not require any synchronization between them. Thus, multiple servers can be located in different places inside the network (see Section 3.3).

3.3 Deployment

Deploying the system inside the ISP network does not require any change in the network configuration or ISP DNS operation. Our system solely relies on protocol listeners and access to ISP network information. Moreover, no installation of special software is required by end-users. The CaTE system adds minimal overhead to ISPs and CDNs. It only requires the installation of one or more CaTE servers in an ISP and the establishment of a connection between ISP and CDN to facilitate communication between them.

Typically, an ISP operates a number of DNS resolvers to better balance the load of DNS requests and to locate DNS servers closer to end-users. To this end, we envision that the ISP's CaTE servers can be co-located with DNS resolvers in order to scale in the same fashion as DNS. CaTE servers can also be located close to peering points in order to reduce the latency between the CDN and an instance of the system. Synchronization of multiple CaTE instances is not necessary as it is implicitly given through the use of protocol listeners. Other possible deployment strategies we have considered are presented in [13].

3.4 Operation

We now describe the operation of our working prototype and its interaction with the CDN. In Figure 2 we illustrate the basic system architecture to support CaTE including the flow of information when the CaTE system is used. When a DNS request is submitted by an end-user to the ISP DNS resolvers *(1)* there are a number of recursive steps *(2)* until the authoritative DNS server is found *(3)*. Then, the ISP DNS resolver contacts the authoritative DNS server *(4)*. There, the request is handed to the content request processor operated by the CDN query processing component *(5)*. The content request processor has access to full information about the status of the CDN. Based on the operational status of the CDN servers, the server selection system [21] is responsible for choosing eligible content servers *(6)*. In the end, a preference list of content servers is generated. At this point, the CDN server selector sends the list of eligible content servers *(A)* along with user information, such as the IP of the DNS resolvers or client and an optimization metric to ISP. The query processor of the ISP system ranks the list using the location ranker *(B)*. After all the elements have been processed, the query processor has an annotated list with preferences for the ISP *(C)*. The query processor sorts the list by the preference values, strips the values and sends the list back to the CDN *(D)*. The CDN server selector incorporates the feedback, selects the best content server(s) and hand them back to the content request processor *(7)*. Then, the answer travels the path back to the client, i.e. from the CDN's authoritative DNS server *(8)* via the ISP DNS resolver *(9)* to the end-user *(10)*. Finally, the end-user contacts the selected server *(11)* and downloads the content *(12)*.

4. EVALUATION

In this section, we quantify the potential benefit of CaTE with different traffic engineering and application goals in mind, namely link utilization, path length, and path delay. For each of these goals we use a separate ranking function, i.e., minimizing the maximum link utilization, reducing the hop-length of the chosen path, and minimizing the delay of the chosen path, respectively. We evaluate CaTE and each ranking function with operational data from a tier-1 ISP and we compare with the performance of current traffic engineering. Recall that CaTE does not change the routing. The comparison with solutions proposed in [8, 16, 30, 31] is out of the scope of this study as they require the change of routing weights. Our analysis is trace-driven and we use our PaDIS emulator [24] to emulate the operation of the ISP and CDNs with and without our CaTE prototype (see Section 3) being deployed.

4.1 Data from a Tier-1 ISP

To build fine-grained traffic demands, we rely on anonymized packet-level traces of residential DSL connections from a tier-1 ISP, henceforth referred to as *ISP1*. For ISP1, we have the complete annotated router-level topology including the routing and its updates, router locations as well as all public/private peerings. ISP1 contains more than 650 routers and 30 peerings around the world. We collect a 10 days long trace starting on May 7, 2010. Our monitor, using Endace monitoring cards [6], observes the traffic of more than 20 000 DSL lines to the Internet. We capture HTTP and DNS traffic using the Bro intrusion detection system [22]. We observe 720 million DNS messages as well as more than 1 billion HTTP requests involving about 1.4 million unique hostnames, representing more than 35 TBytes of data. More than 97% of all DNS requests are using DNS resolvers of ISP1. With regards to the application mix, more than 65% of the traffic volume is due to HTTP.

A significant fraction of the Internet traffic is due to large CDNs. We rank the CDNs by decreasing traffic volume. The top 10 CDNs

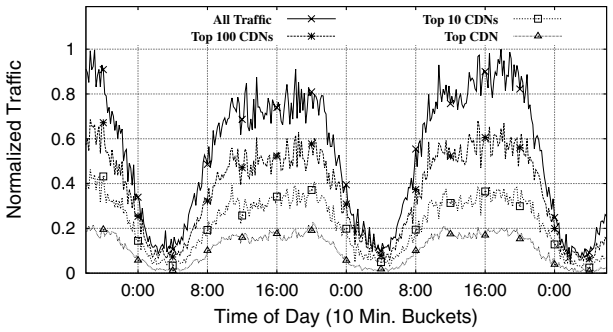

Figure 3: Normalized traffic for top CDNs by volume in ISP1.

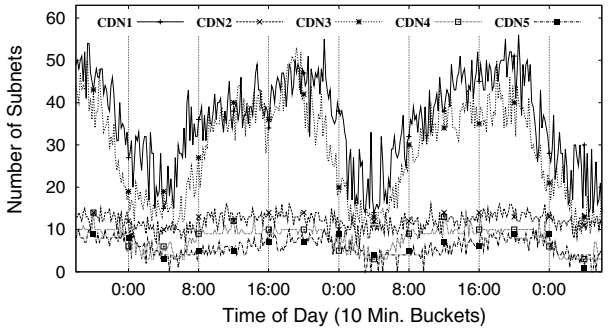

Figure 4: Number of subnets for selected top CDNs.

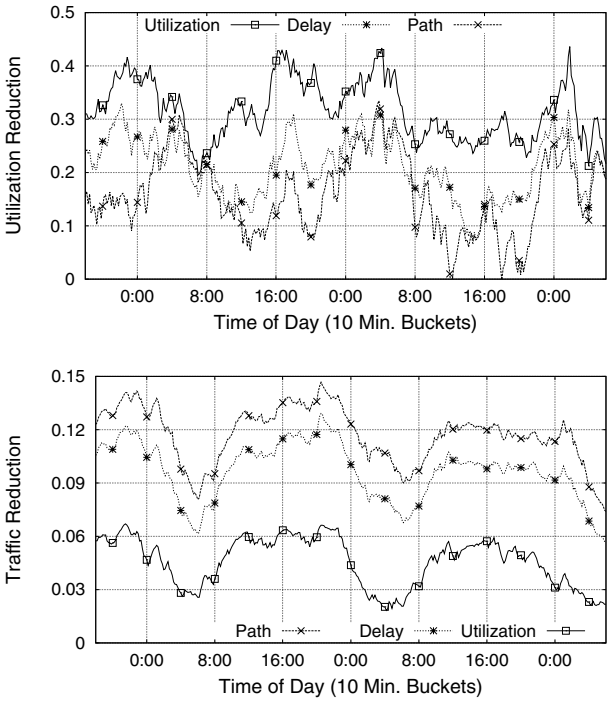

Figure 5: Maximum link utilization (top) and total traffic (bottom) reduction with CaTE and different ranking functions.

are responsible for around 50% of the HTTP traffic volume while the top 100 CDNs for 70% during the peak hour as shown in Figure 3. The marginal increase of traffic is diminishing when increasing the number of CDNs. This shows that collaborating directly with a small number of large CDNs would enable the engineering of a significant fraction of the traffic.

4.2 Location Diversity of CDNs

We examine the location diversity of CDNs based on the traces from ISP1. Figure 4 shows the number of exposed subnets of five of the top 10 CPs by volume. We observe that 50% of the HTTP traffic can be delivered from servers in at least 8 different subnets, and more than 60% comes from more than 3 different subnets. When only looking at the top 10 CDNs by volume, we find that the content can typically be served from a large number of locations (tens of subnets). We also notice that the diversity exposed by some CDNs exhibits time of day patterns, while others do not. We attribute this to the CDNs deployment scheme. To exemplify, a large CDN that installs servers deep inside the ISPs can redirect users to a large number of locations, especially during the peak hour. On the other hand, a datacenter-based CDN that deploys servers in strategic locations tends to send users repeatedly to the same locations.

4.3 CaTE in ISP1

CaTE allows ISPs and CDNs to utilize different ranking functions: minimization of the maximum link utilization ("Utilization"), minimization of the hop-length of the chosen path ("Path"), and minimization of the delay of the chosen path ("Delay"), to optimize for link utilization, path length, and delay respectively. We quantify the effects of CaTE when using each one of these individually. We focus on the top 10 CDNs, as these have a significant share of the overall traffic. Similar observations are made when applying CaTE to the top 1 and 100 CDNs [13]. To evaluate the effectiveness of CaTE, we observe the network traffic and measure network quantities such as maximum link utilization as the CDNs apply their server selection algorithms. We then assume that the CDNs follows the recommendations provided by the ISP, and we estimate the effects on these network quantities.

Effect on maximum link utilization: In Figure 5 (top) we plot the maximum link utilization reduction when applying CaTE to the top 10 CDNs with different optimization goals. The first observation is that a significant reduction occurs for all optimization goals. When optimizing for link utilization, the reduction is the highest, up to 40%. Even when optimizing for delay or path length, two optimizations that do not target the reduction of maximum link utilization, the reduction can be up to 30%.

Effect on network-wide traffic: In Figure 5 (bottom) we plot the network-wide traffic reduction. Again, the traffic reduction is significant for all three optimization goals. We also observe a diurnal pattern in the network-wide traffic reduction, corresponding to the diurnal pattern in the demand. When optimizing for path length, the traffic reduction in the network is the highest, up to 15% in the peak hour. This is to be expected as less traffic traverses long paths.

Effect on path length: To quantify the path length reduction in ISP1 with different optimization goals, we plot the relative traffic across different path lengths inside the network in Figure 6 (top). CaTE redirects the traffic towards paths with the same or even shorter length. Notice that there is no traffic for path length equal to 1 due to the network design in ISP1. The most significant shift is achieved when optimizing for path length, but major improvements are also achieved when delay or link utilization are used.

Effect on path delay: Figure 6 (bottom) shows the accumulated path delay for the traffic ISP1 when using CaTE with different optimization goals. As expected, the highest improvement in the path delay is achieved when path delay is minimized. Significant improvements are also achieved when the optimization goal is path length and link utilization, especially for high values of path delays.

Active measurements: We complement our network-wide emulation with active measurements. For this, we focus on a major CDN that is responsible for more than 8% of the overall traffic.

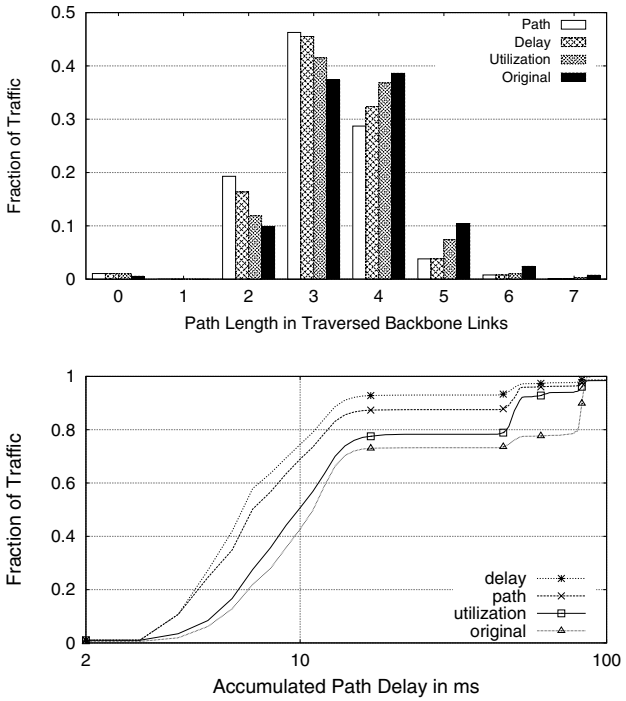

Figure 6: Backbone path length (top) and accumulated path delay (bottom) with CaTE and different ranking functions.

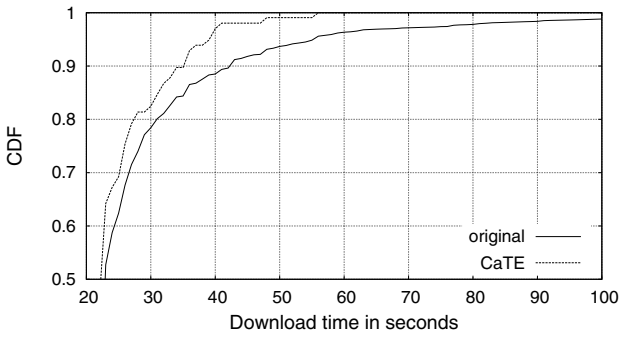

Figure 7: Distribution of download times of a CDN.

This CDN is an OCH distributed across 12 locations. Over a period of one week, we repeatedly downloaded a 60 MBytes object from this CDN. The downloads were performed every two hours for each of the 12 locations. Additionally, mapping requests were issued every 200ms to find out the dynamics in the server assignment of this CDN. Figure 7 shows the distribution of total download times when the CDN assigns end-users to its servers ("original") and compares it to the possible download times that when CaTE is used. We observe that more than 50% of the downloads do not show a significant difference. This happens mainly during non-peak hours. For 20% of the downloads, we observe a significant difference in the download times, mainly during peak hours. This confirms our observation that CaTE is most beneficial during peak hours.

4.4 CaTE and Popular Applications

One of todays biggest challenges for ISPs is dealing with rapid shifts of high traffic volumes, as currently deployed traffic engineering tools are too slow to react in a timely fashion. For example, Netflix, a very popular application delivering high quality videos to end-users, relies on commercial CDNs. Recent studies show that

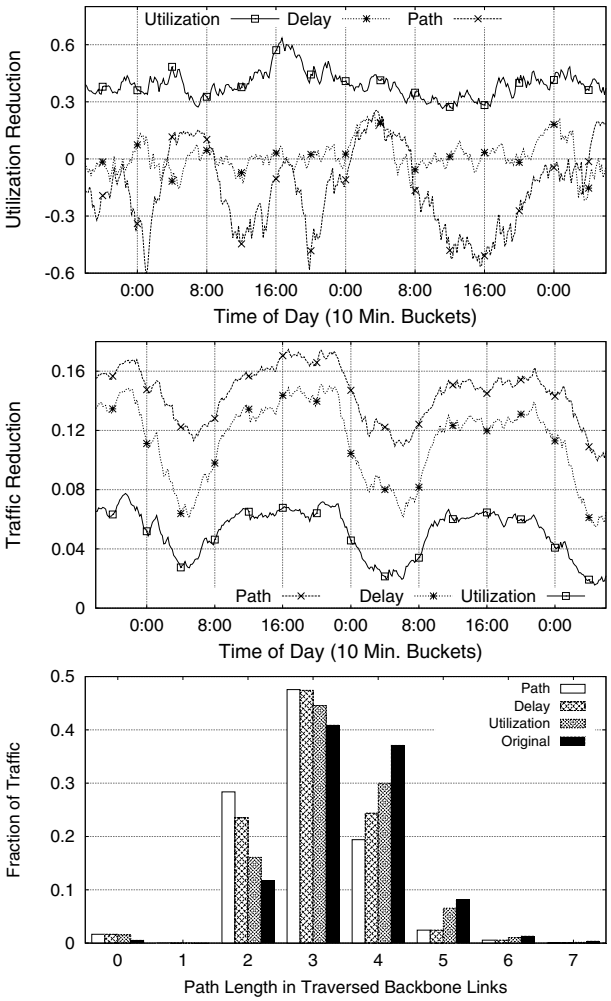

Figure 8: Netflix in ISP1: Projection of the reduction in link utilization (top) and overall network traffic (middle), and volume percentage by path length (bottom).

Netflix is responsible for more than 30% of the peak downstream traffic in large ISPs [15]. While only available in North and Latin America today, Netflix has announced the launch of its service in Europe in 2012. To quantify the effect of Netflix being deployed in Europe, we assume that one of the CDNs currently hosting Netflix increases its traffic 20-fold (with the same request distribution), and redo the analysis from the previous section. We note that a major part of ISP1 footprint and customer base is in Europe.

Our results show that with CaTE, the utilization of the most utilized link can be reduced by up to 60% (see top of Figure 8), the total HTTP traffic volume can be reduced by 15% (see middle of Figure 8) and traffic can be shifted towards shorter paths inside the network of ISP1 (bottom of Figure 8). However, when considering all metrics, we observe that not all metrics can be optimized to their full extent simultaneously. For example, a reduction of traffic on the order of 15% would actually increase the utilization on the highest loaded link by 60%. This indicates that the optimization function employed by CaTE needs to be carefully chosen to target the most important metrics when deploying CaTE inside a network. Nonetheless, if CaTE optimizes for reduction of the maximum link utilization, benefits in all metrics can be observed.

Popular Internet applications such as Netflix deploy their infrastructure in order to improve end-user experience and peering with

ISPs. Recently, Netflix decided to roll out its own CDN and allow ISPs to get the Netflix data at common interest exchanges [1]. CaTE can be utilized to locate the best exchange points between Netflix servers and the ISP.

5. RELATED WORK

The closest work to ours is a game-theoretic study [16]. The focus is on the joint optimization of traffic engineering inside an ISP and server selection by a CDN that is operated by the same ISP. The authors prove that the exchange of information is mutually beneficial. However they do not propose a system to realize this. They do not also specify a protocol where the CDN communicates the set of candidate servers and an ISP ranks them as they rely on a centralized solution. This exchange protocol is desirable when the two parties are not under the same administration as we showed in our evaluation section. In [30] the authors address two shortcomings of [16], namely the centralized collaboration protocol and the collaboration between only one ISP and one CDN. In [8] the authors prove the existence of equilibria where both traffic engineering goals and improved server selection are possible if ISP signals paths as congested. A recent study [26] shows that the utilization of path diversity offered by massively distributed CDN and hosting infrastructures can be beneficial for ISPs. P4P [31] and IETF ALTO working group [4] have proposed protocols where ISPs can provide partial or fully annotated topology or distance information to the end-users and applications. P4P focuses on the reduction of ISP interdomain traffic cost and it provides an interface for peer-to-peer applications and users to communicate with network providers to get the state of their network. ALTO considers the interaction between ISPs and CDNs as well as applications. We explain how we address some of the shortcomings of these approaches in our system design in Section 3.2. IETF CDNi working group [5] is also related. The focus is on the interconnection of CDNs to reduce operating and capital costs.

6. SUMMARY AND FUTURE WORK

In this paper we introduce and evaluate CaTE, a new concept for traffic engineering that leverages the fact that a significant fraction of content demand can be served from multiple locations. CaTE relies on the observation that by selecting appropriate servers, the path of the traffic in the network can be influenced in a desired way. We show that CDN and ISP collaboration only in server selection, not routing, is sufficient for traffic engineering and end-user experience improvement. This is a clear differentiator from previous works on CDN-ISP cooperation that focus on routing.

To enable CaTE, we design, implement and evaluate a prototype system that can scale up to thousands of requests per second (per instance of the system) thanks to its novel design on how to gather and maintain network data, as well as the interaction between the proposed ISP and existing CDN query processing systems. Our experimental results using traces from a large tier-1 ISP and a number of CDNs show that CaTE can provide performance benefits for CDNs, ISPs, and end-users under a number of network metrics.

To capitalize on the substantial performance benefits that CaTE offers to CDNs, ISPs and end-users we are currently investigating how to install CaTE in a number of ISPs and enable collaboration with CDNs. We would also like to quantify the effect of ISP topology and CDN deployment on the effectiveness of CaTE.

7. REFERENCES

[1] Announcing the Netflix Open Connect Network. http://blog.netflix.com/2012/06/announcing-netflix-open-connect-network.html.

[2] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing DNS Resolvers in the Wild. In *ACM IMC*, 2010.

[3] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Web Content Cartography. In *ACM IMC*, 2011.

[4] R. Alimi, R. Penno, and Y. R. Yang. draft-ietf-alto-protocol-08, 2011.

[5] IETF CDNi. draft-ietf-alto-protocol-08, 2012.

[6] J. Cleary, S. Donnelly, I. Graham, A. McGregor, and M. Pearson. Design Principles for Accurate Passive Measurement. In *PAM*, 2000.

[7] C. Contavalli, W. van der Gaast, S. Leach, and D. Rodden. Client subnet in DNS requests. draft-vandergaast-edns-client-subnet-01.

[8] D. DiPalantino and R. Johari. Traffic Engineering vs. Content Distribution: A Game-theoretic Perspective. In *IEEE INFOCOM*, 2009.

[9] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE : MPLS adaptive traffic engineering. In *IEEE INFOCOM*, 2001.

[10] S. Fischer, N. Kammenhuber, and A. Feldmann. REPLEX: Dynamic Traffic Engineering based on Wardrop Routing Policies. In *ACM CoNEXT*, 2006.

[11] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS Weights in a Changing World. *IEEE J. Sel. Areas in Commun.*, 20(4), 2002.

[12] P. Francois, M. Shand, and O. Bonaventure. Disruption Free Topology Reconfiguration in OSPF Networks. In *IEEE INFOCOM*, 2007.

[13] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann. Content-aware Traffic Engineering. *CoRR*, abs/1202.1464, 2012.

[14] A. Gerber and R. Doverspike. Traffic Types and Growth in Backbone Networks. In *OFC/NFOEC*, 2011.

[15] Sandvine Inc. Global broadband phenomena, 2011.

[16] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. Cooperative Content Distribution and Traffic Engineering in an ISP Network. In *ACM SIGMETRICS*, 2009.

[17] R. Krishnan, H. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving Beyond End-to-end Path Information to Optimize CDN Performance. In *ACM IMC*, 2009.

[18] C. Labovitz, S. Lekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-Domain Traffic. In *ACM SIGCOMM*, 2010.

[19] T. Leighton. Improving Performance on the Internet. *Communications of the ACM*, 52(2), 2009.

[20] Z. Mao, C. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A Precise and Efficient Evaluation of the Proximity Between Web Clients and Their Local DNS Servers. In *USENIX Annual Technical Conference*, 2002.

[21] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai Network. *SIGOPS Rev.*, 44(3), 2010.

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

[23] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving Content Delivery using Provider-aided Distance Information. In *ACM IMC*, 2010.

[24] I. Poese, B. Frank, S. Knight, N. Semmler, and G. Smaragdakis. PaDIS Emulator: An Emulator to Evaluate CDN-ISP Collaboration. In *ACM SIGCOMM*, 2012.

[25] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the Electric Bill for Internet-scale Systems. In *ACM SIGCOMM*, 2009.

[26] A. Sharma, A. Mishra, V. Kumar, and A. Venkataramani. Beyond MLU: An application-centric comparison of traffic engineering schemes. In *IEEE INFOCOM*, 2011.

[27] S.Kandula, D. Katabi, B. Davie, and A. Charny. Walking the Tightrope: Responsive Yet Stable Traffic Engineering. In *ACM SIGCOMM*, 2005.

[28] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering What-if Deployment and Configuration Questions with Wise. In *ACM SIGCOMM*, 2009.

[29] S. Triukose, Z. Al-Qudah, and M. Rabinovich. Content Delivery Networks: Protection or Threat? In *ESORICS*, 2009.

[30] P. Xia, S.-H. G. Chan, M. Chiang, G. Shui, H. Zhang, L. Wen, and Z. Yan. Distributed Joint Optimization of Traffic Engineering and Server Selection. In *IEEE Packet Video Workshop*, 2010.

[31] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz. P4P: Provider Portal for Applications. In *ACM SIGCOMM*, 2008.