Technische Universität Berlin
Fakultät für Elektrotechnik und Informatik
Lehrstuhl für Intelligente Netze
und Management Verteilter Systeme

Towards Collaborative Internet Content Delivery

vorgelegt von

Ingmar Poese (Dipl.-Inf.)

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Jean-Pierre Seifert, Technische Universität Berlin, Germany
Gutachterin:	Prof. Anja Feldmann, Ph.D., Technische Universität Berlin, Germany
Gutachter:	Prof. Bruce Maggs, Ph.D., Duke University, NC, USA
Gutachter:	Prof. Dr. Steve Uhlig, Queen Mary, University of London, UK
Gutachter:	Georgios Smaragdakis, Ph.D., Technische Universität Berlin, Germany

Tag der Wissenschaftlichen Aussprache: 22 April 2013

Berlin 2013
D 83

Ich versichere von Eides statt, dass ich diese Dissertation selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Datum Ingmar Poese

Abstract

Today, a large fraction of Internet traffic is originated by Content Distribution Infrastructures (CDIs), such as content distribution networks, hyper-giants and One-Click-Hosters. To cope with the increasing demand for content, CDIs deploy massive centralized or distributed infrastructures. For CDIs, the operation of their infrastructures is challenging, as they have to dynamically map end-users to appropriate servers without being fully aware of the end-users' network locations. Apart from CDIs, the operational overhead of Internet Service Providers (ISPs) is growing increasingly complex, due to content delivery traffic caused by CDIs. In fact, the difficulties ISPs have with regards to engineering their traffic, stem from the fact that CDIs have limited knowledge about network conditions and infrastructures, while ISPs cannot communicate their insight about networks to CDIs.

To solve the mapping challenges CDIs face, we studying the applicability of IP-Geolocation to optimize CDI operation in terms of end-user to server mapping. We base the study on two different approaches: a) an evaluation of end-user submitted GPS coordinates and b) a general study of IP-Geolocation databases. We find that in both cases, IP-Geolocation is only of limited help to select servers close to end-users. Especially in mobile environments, we find that IP-Geolocation is unable to solve the mapping problem.

We broaden the scope and tackle CDIs' general lack of awareness with regards to ISP networks. We argue that the challenges CDIs and ISPs face today can be turned into an opportunity when enabling collaboration between the two. We propose, design and implement a solution, where an ISP offers a *Provider-aided Distance Information System* (PaDIS) as an interface for CDIs. PaDIS uses information available only to the ISP to rank any client-host pair, based on up-to-date network information, such as delay, bandwidth or number of hops. By extension, this approach also implicitly solves the mapping problem IP-Geolocation was unable to resolve. Experiments with different CDIs show that improvements in download times of up to a factor of four are possible. Furthermore, we show that deploying PaDIS not only benefits CDIs, but also end-users.

With regards to the benefits for ISPs, we show that by offering PaDIS to CDIs, ISPs are able to partly reclaim control of the traffic induced by CDIs. We design the concept of *Content-aware Traffic Engineering* (CaTE), which dynamically adapts the traffic demand for content hosted on CDIs by utilizing PaDIS during their server selection process. As a result, CDIs enhance their end-user to server mapping and improve end-user experience. In addition, ISPs gain the ability to partially influence traffic demands within their networks. Our evaluation, based upon operational data from a large tier-1 ISP, shows improvements minimizing the path length as well as delay between end-user and assigned CDI server, significant reduction in network-wide traffic and in maximum link utilization.

Zusammenfassung

Der heutige Internetverkehr wird von *Content Distribution Infrastructures (CDI)*, z.B. Content Distribution Networks, Hyper-Giants und One-Click-Hostern, dominiert. Um das rasante Wachstum der Daten zu bewältigen, betreiben CDIs massive Infrastrukturen. Aber mit deren Größe wachsen auch die operativen Herausforderungen. Hier erweist es sich als schwierig, die Server-zu-User Zuweisung aktuell zu halten, da dem CDI die Netzwerktopologie, deren aktueller Zustand und die genaue Netzwerkposition des Users nicht bekannt sind. Zur gleichen Zeit sehen sich Netzwerkbetreiber, genannt Internet Service Provider (ISP), mit dem stark wachsenden und immer unberechenbarer werdenden Verkehrsverhalten der CDIs ausgesetzt, ohne darauf entsprechend reagieren zu können. Diese Schwierigkeiten zwischen ISPs und CDIs resultieren aus dem fehlenden Mechanismus der ISPs ihr Netzwerkwissen an die CDIs zu kommunizieren.

Um das Server-zu-User Zuweisungsproblem zu lösen, beschäftigen wir uns zuerst mit IP-Geolocation. Hier legen wir den Fokus auf zwei Ansätze: a) das Auswerten von GPS Koordinaten, die von Usern selbst bereitgestellt werden und b) eine generelle Studie von IP-Geolocation Datenbanken. In beiden Fällen wird deutlich, dass IP-Geolocation nur sehr begrenzt helfen kann. Besonders in mobilen Datennetzen ist IP-Geolocation keine Hilfe.

Als nächstes wenden wir uns dem Problem der Netzwerk- und Topologieunkenntnis von CDIs zu. Unsere Lösung dieses Problems beruht auf Kooperation zwischen CDIs und ISPs. Hierzu beschreiben, entwickeln und implementieren wir das *Provider-aided Distance Information System* (PaDIS), welches von ISPs betrieben wird und es CDIs ermöglicht, aktuelle Netzwerkinformationen zu beziehen. Dazu benutzt PaDIS Information aus dem operativen Betrieb einer ISP um CDIs den besten Server, basierend auf verschiedenen Metriken, wie Auslastung, Hops oder Latenz, vorzuschlagen. Außerdem wird durch PaDIS auch das Server-zu-User Zuweisungsproblem gelöst, was mit IP-Geolocation nicht möglich war. Unsere Auswertung zeigt dabei, dass PaDIS die Zeiten zum Herunterladen von Dateien um einen Faktor von Vier verkürzen kann. Davon profitieren nicht nur CDIs, sondern auch die User.

ISPs ziehen aus dem Einsatz von PaDIS den Vorteil, dass sie die Zuweisung von Server-zu-User mitsteuern können. Wir entwerfen das Konzept des *Content-aware Traffic Engineering* (CaTE), welches den Verkehr von CDIs dynamisch an die aktuelle Last von Netzwerken anpasst. Im Ergebnis wird die Zuordnung von Server-zu-User deutlich verbessert, was sich sowohl positiv für das CDI als auch die User auswirkt. Weiterhin erlangen ISP die Fähigkeit, Datenströme auf Netzwerkpfade mit wenig Belastung zu legen. Unsere Auswertung von CaTE, welche auf operativen Daten einer ISP beruht, zeigt, dass sowohl die Pfadlängen als auch die Latenz zwischen Server und User signifikant verringert werden, während die ISPs ihren Datenverkehr gleichmäßiger verteilen können und dadurch die Gesamtlast des Netzwerks senken.

Pre-published Papers

Parts of this thesis are based on the following papers that have already been published. All my collaborators are among my co-authors.

International Conferences

Ingmar Poese, Bejamin Frank, Bernhard Ager, Georgios Smaragdakis, Anja Feldmann. **Improving Content Delivery using Provider-aided Distance Information**. In Proceedings of Internet Measurement Conference (IMC). Melbourne, Australia. November 2010.

International Journals

Benjamin Frank, Ingmar Poese, Yin Lin, Rick Weber, Janis Rake, Georgios Smaragdakis, Anja Feldmann, Steve Uhlig, Bruce Maggs **Pushing ISP-CDN Collaboration to the Limit.** In ACM SIGCOMM Computer Communication Review (CCR). Currently under Submission.

Ingmar Poese, Benjamin Frank, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann, Bruce Maggs **Enabling Content-aware Traffic Engineering**. In ACM SIGCOMM Computer Communication Review (CCR). October 2012

Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann **Improving Content Delivery with PaDIS**. In IEEE Internet Computing. May-June 2012.

Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, Bamba Gueye. **IP Geolocation Databases: Unreliable?.** In ACM SIGCOMM Computer Communication Review (CCR). April 2011.

Posters and Demos

Ingmar Poese, Benjamin Frank, Simon Knight, Niklas Semmler, Georgios Smaragdakis. **PaDIS Emulator: An Emulator to Evaluate CDN-ISP Collaboration.** In Proceedings of ACM SIGCOMM 2012 (demo session). Helsinki, Finland. August 2012.

Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann. **Content-aware Traffic Engineering.** In Proceedings of ACM SIGMETRICS 2012 (poster session). London, United Kingdom. July 2012.

Technical Reports

Benjamin Frank, Ingmar Poese, Georgios Smaragdakis, Steve Uhlig, Anja Feldmann **Content-aware Traffice Engineering.** Technical Report TU-Berlin. Berlin, Germany. February 2012.

Ingmar Poese, Mohamed Ali Kaafar, Benoit Donnet, Bamba Gueye, Steve Uhlig **IP Geolocation Databases: Unreliable?.** Technical Report TU-Berlin. Berlin, Germany. April 2011.

Contents

1 Introduction

User demand for popular applications, including online gaming, music, and video (e.g., YouTube and IPTV), is phenomenal and expected to grow [26]. As a result, satisfying user demands requires continuous upgrades to both networks and large-scale service infrastructures. Currently, Content Distribution Infrastructures (CDIs), such as content delivery networks (CDNs), One-Click Hosters (OCH), Video Stream Portals (OCH) and Online Social Networks (OSN), answer the increasing demand for their services and richer content by deploying massive hosting infrastructures. But the increase in deployed infrastructures and its operation is (a) costly: installation and basic operational costs have to be paid for the contract period even if the servers are under-utilized, and (b) time consuming (multiple months to years): hardware has to be physically installed and operated. Once installed, current hosting and peering contracts typically last for months or years [109], making provisioning for peak demand at an unknown future scale prohibitively expensive.

Most applications hosted on CDI infrastructures rely on direct interactions with end-users and are very sensitive to delay [85]. Indeed, transaction delay is critical for online businesses to succeed [77]. Here, network delay and loss are important contributors. Thus, CDIs strive to locate their servers close to - in terms of network distance - end-users. For example, CDIs implement sophisticated methods to infer network conditions to improve perceived end-user experience [102] through active measurements within ISPs. Yet, discovering and maintaining a view on networks help CDIs to generally achieve a reasonable performance, but also induces significant analysis overhead and frequent mismatches in the mapping [107]. Besides keeping track of the network state, CDIs also need information on the location of users in the network in order to choose close-by servers.

CDIs mainly rely on DNS when mapping end-users to servers. However, due to the inability of DNS to communicate the original source of the request, CDIs loose all information on where an end-user is located, and instead only obtain information on the DNS resolver that is being used. This leads to the assumption that DNS resolvers used by end-users are close to them. However, it has been shown that this is not always the case [6]. The emergence of third party DNS resolver further complicates the mapping process, since even the ISP that originates the request cannot be determined reliably anymore.

There have been several proposals to overcome this challenge. One of the most obvious solutions to solve the challenge of selecting a close server for an end-user is IP-Geolocation. By choosing a server in the same city the chances are high that the network path is also short. However, the network and the geographical distance between two points on the Internet are not necessarily related. Thus, an approach based on IP-Geolocation cannot stand alone, but needs to be combined with information gleaned from network operation to be successful.

A promising proposal to improve network proximity is the combination of a DNS [32] extension (EDNS0) combined with the ALTO topology discovery service ALTO [119]. The DNS extension is currently discussed at the IETF. It basically enables DNS resolvers to communicate the origin of a request. This, in turn, allows CDIs to get information about the end-users network location in terms of an IP address or subnet. Also, EDNS0 mitigates the effects third-party DNS resolvers have on server to end-user mapping due to the added network information included in the request. In order to alleviate CDIs from the burden of topology inference, the IETF ALTO group is currently working on a protocol allowing CDIs to request topological information from ISPs. Using both proposals together enables CDIs to take the infrastructure topology into account while effectively finding the network location of end-users and thus improving end-user to server mapping. However, a CDI using the topological information obtained through ALTO can better match its operation to the infrastructure, but has no information on the operation of other CDIs also operating in the ISP. Furthermore, ALTO does not allow for dynamic information to be conveyed, i. e., link utilizations, delays or short term failures. Thus, the combination of EDNS0 and ALTO solves the mapping and the topological shortcomings of CDI operation. At the same time, it does not alleviate them from needing to keep track of link over-utilization, network failures and route changes.

All information CDIs are missing today to optimize their operation is already available to the ISP. Through ALTO, ISPs have a channel that, in theory, allows them to partially share that information with CDIs. However, this comes at the cost of (a) sharing potentially business critical topological information with CDIs, (b) little to no operational improvement for the ISP and (c) the inability to include dynamic operational information. Thus, with incentives and operational improvement ISPs are reluctant to implement the needed services.

1.1 Building a Collaborative Content Delivery System

We tackle the problem of optimizing CDI operation by identifying the information that each party needs while minimizing the information exposure. However, no such system exists today. We argue that a CDI does neither need nor want topological information of an ISP, while ISPs have no expertise or intention of optimizing CDI operation. To this end, *we design, implement and evaluate the* ***Provider-aided Distance Information System (PaDIS)***, based on the work of Aggrawal et al. [9, 10] on P2P-ISP collaboration. PaDIS is run by an ISP to offer CDIs exactly the information that is needed to improve their operation without revealing operation secrets. It allows an arbitrary number of CDIs to collaborate with an ISP and obtain mapping and operation information based on to the network infrastructure without the need to know them. Through this, CDIs improve their end-user to server mapping, are relieved of extensive network measurements and improve their customer satisfaction through accelerated content delivery. In technical terms, this means that the selection of eligible servers to answer a client request are assembled by a function that only the CDI knows. This list is then sent to the ISP's PaDIS, where it is rearranged in terms of the network path between servers supplied by the CDI and the client that sent the request. In the end, only the rearranged list is sent back.

To accelerate the deployment of PaDIS, ISPs also needed incentives. Only if both, ISPs and CDIs, benefit from PaDIS can it succeed in improving content delivery. Thus, *we propose, design, model, implement and evaluate the concept of* ***Content-aware Traffic Engineering (CaTE)****, which is based on PaDIS* . In a nutshell, CaTE leverages the observation that, by choosing a server location for a CDI, a network path is also chosen. Thus, by selecting the server in such a way, not only the CDI is optimized, but also the traffic induced by the CDI is put on a desirable network path. At the same time, we make sure that the improvements the CDI gains by collaboration through PaDIS are maintained. Also, our design ensures that CaTE is complementary to today's traffic engineering schemes and does not induce feedback loops which lead to network instabilities. In fact, CaTE aims at engineering traffic at short time scales, i. e., in the order of minutes, whereas today's traffic engineering works on traffic aggregates and operates in time scales of days, if not weeks. Likewise, CaTE does not interfere with protocol based flow control mechanism, i. e., TCP, as it leaves these individual connection untouched.

Finally, *we sketch the concept of* ***Network Platform as a Service (NetPaaS)***. NetPaaS is built upon PaDIS and CaTE, but it brings the notion of an ISP-operated cloud infrastructure, called Microdatacenter, to the discussion. In fact, if CDIs and ISPs already jointly optimize their operation, a natural extension is to allow the CDI to directly deploy servers within an ISP. Thanks to the commoditization of hardware virtualization, this would be a straightforward task for ISPs. With NetPaaS, todays massive scale content distribution can be significantly improved.

1.2 Methodology and Verification

Introducing new concepts and systems to the Internet landscape raises the question of how and what needs to be changed or adapted, as well as quantifying the impact on the involved parties and the Internet. PaDIS, and the concepts built upon it, have the potentially to affect billions of requests each day. Thus, it is important to design and evaluate the system thoroughly.

We base all evaluation either on operational data from ISPs and CDIs, or conduct extensive active and passive measurements to verify our system design. We evaluate the need for PaDIS based on operational traces from both, ISPs and CDIs. When turning to the applicability as well as the effects of PaDIS on CDIs, we use passive packet level traces and well as active measurements with multiple CDIs. Furthermore, we perform stress tests on our PaDIS prototype to show that it (a) manages the vast information available in ISP networks with ease and (b) is up to the task of scaling to the performance needed to run in operational networks. Turning to the results of the PaDIS evaluation, we find that the delivery of small objects from a CDI can be improved by a factor of two. Furthermore, we show that PaDIS is capable of reducing download times of popular One-Click Hoster (OCH) to a quarter at peak hours in a targeted ISP.

In the case of CaTE, we introduce the theoretical foundation of why and how it works and design a framework to simulate its effects. This is needed as CaTE is designed to run in large networks with massive traffic which cannot be recorded or replayed at sufficient scale. Thus, we carefully model the traffic in a simulation that is based again on operational data obtained from an ISP and a CDN. Our evaluation of CaTE shows that the network-wide traffic volume in the evaluated ISP is reduced up to 6%, while the maximum link utilization drops by more than 50%.

1.3 Outline

The rest of the thesis is structured as follows: In Section 2 we introduce the background of ISPs and CDIs operation. In Section 3 we analyze the the operations of CDIs in an ISP. Section 4 focuses on IP-Geolocation in regard to its accuracy and ability to help CDIs in terms of server to user mapping.

Section 5 introduces PaDIS, explains its architectural designs, goals and how it fits into todays content distribution ecosystem. Section 6 introduces the concept of CaTE, by modeling the theoretical approach and evaluating it based on operational data. Finally, in Section 7, our future work regarding NetPaaS is sketched.

Section 8 relates previous and current work to this thesis while Section 9 concludes the work on collaborative Internet content delivery.

1.4 Contributions

The contributions of this thesis are as follows:

CDI mapping, behavior and identification :

Based on passive and active Internet network measurements, as well as operational logs from a large CDN we show that

- CDIs can be identified through their DNS-based mapping strategies
- CDIs indeed operate large scale distributed infrastructures
- CDI end-user to server mapping can be improved in performance
- the deployment of CDIs is not optimal

Geolocation reliability:

The analysis of user submitted geolocation as well as IP-Geolocation databases shows that IP-Geolocation cannot be relied on to improve the CDI's operation and performance.

Provider-aided Distance Information System (PaDIS):

We present PaDIS to enable ISPs to aid CDIs in their server selection. We introduce the information requirements, the processing as well as the necessary abstractions needed to enable scalable augmented server selection of CDIs and develop the mechanism to transparently perform CDI-ISP collaboration for the existing CDI and ISP infrastructures of today's Internet.

Content-aware Traffic Engineering (CaTE):

We introduce the concept of CaTE for near-real time traffic engineering of ISPs based on collaboration between CDIs and ISPs. The proposed algorithms for solving CaTE as well as their fast heuristics allow for a new kind of network traffic management for ISPs, which complements todays traffic engineering toolbox while giving partial control of traffic back to ISPs.

2 Background

Internet traffic grows at a rate of approximately 30% per year [26] and is dominated by the delivery of content to end users [5, 54, 80, 107]. To cope with the increasing demand for content, and to support the level of reliability and scalability required by commercial-grade applications, Content Distribution Infrastructures (CDIs) have emerged. Basically, a CDI is an infrastructure to distribute and deliver content for itself or for third parties. CDIs include, but are not limited to, Content Distribution Networks (CDNs), such as Akamai and Google, Video Streaming Portals (VSP) such as Youtube, One-Click-Hosters (OCH) like Rapidshare and Fileserve. Likewise, we define a Content Producer (CP) as the entity that generates content as opposed to delivering it. In some cases, e. g., Google and Youtube, the CP and CDI can be the same entity. In other instances, for example Akamai and Limelight, a CDI only delivers what a CP pays for. To better understand how content delivery works on the Internet, we first give a high level overview on how the infrastructure and operation of the Internet is formed and maintained by Internet Service Providers (ISPs). Then we turn to the concept of CDIs, explain their role in today's Internet and highlight CDI architecture currently in use as well as trends for the future.

2.1 Internet Service Providers (ISPs)

An Internet Service Provider (ISP) is, in very general terms, an entity that operates a network for connecting end-users, company and organizations to remote, infrastructures or other ISPs. The Internet is formed by the interconnection of multiple individual networks run by ISPs. However, control of an individual network remains solely with the ISP operating it. Figure 2.1 shows how the Internet is formed.

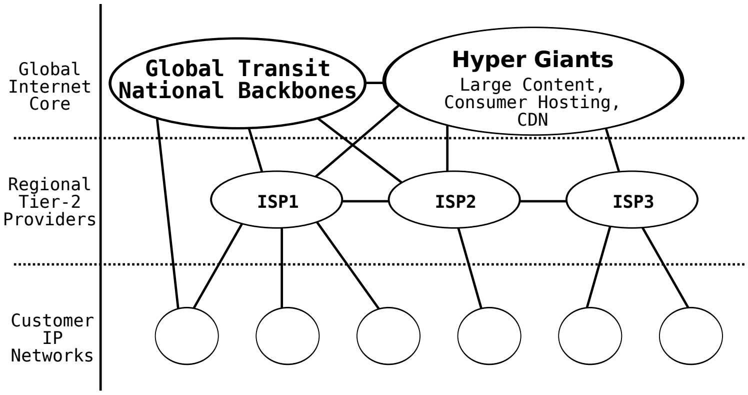

Global Internet Core
Global Transit National Backbones
Hyper Giants
Large Content, Consumer Hosting, CDN
Regional Tier-2 Providers
ISP1
ISP2
ISP3
Customer IP Networks

Figure 2.1: **Layout of the Internet Structure**

Here, the ISPs run their own networks. This forces a clear distinction between the individual network that an ISP runs and the global Internet as a network of networks. Also, from this, it can be deduced that nobody has control over the Internet, but instead each ISP has only control over its own network and the direct connections to other networks.

Two schemes are fundamental to the operations and scalability of today's Internet. The first scheme is related to the administration of a single network run by an ISP, called an Autonomous System (AS). An ISP runs at least one AS, but can choose to run more than one if it wants to partition its network. Each AS is, in general, under the control of one administrative domain. Each AS is usually managed by an Interior Gateway Protocol (IGP), e. g., OSPF [96] or ISIS [103]. Since an AS is run centrally by one instance, there is no need for information aggregation and/or hiding. Thus, each member of an AS can have full topological and operational knowledge of the entire AS. Unless otherwise stated, we assume that one ISP runs exactly one AS.

The second scheme regards the interconnection of different ASes. The idea is simple: ASes need to be interconnected in order to be able to communicate with each other and form the Internet. To keep the information transported in the communication scalable throughout the Internet, the entire internal management of the individual AS is abstracted and aggregated. In general terms, ASes only exchange reachability information, i. e., they only tell each other what part of the Internet they can reach, but not how. Furthermore, there are cases when an AS needs to communicate with another AS that it does not have a direct connection to. In this case, the communication has to transit one or more different ASes. Thus, along with with the pure reachability information, the AS number (which is unique for each AS) is also transmitted. This allows for loop detection as well as an estimate of how many AS hops away a destination is. Today, the Border Gateway Protocol (BGP [110]) is the de-facto standard employed when one AS communicates with another AS.

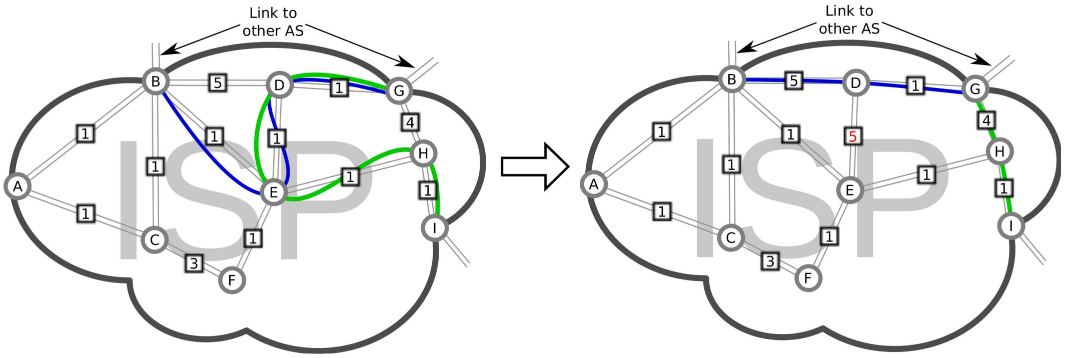


Figure 2.2: **IGP based Traffic Management Example**

Returning to Figure 2.1, we also observe that ISPs have positioned themselves with different objectives to best suit their needs. This can be due to the size of the network managed by an AS as well as the focus the ISP puts towards the offered services. For example, a regional ISP offering subscriptions to end-users neither needs nor wants to run a global backbone. Instead, it can just buy global connectivity from an ISP that is specialized in providing these services. This leads to a hierarchical structure of the Internet architecture. However, the hierarchy has undergone drastic changes in the past years [5, 80]. The changes are driven by Hyper-giants, i. e., massive Content Distributors that break with the traditional, hierarchical structure of the Internet. Their business is to deliver content as quickly as possible. This necessitates bringing their services as close to end-users as possible, and the best way to do this is to create connections with as many ASes as possible, or to deploy their services directly to the ISPs' networks. At the same time, they do not function like traditional transit providers. Thus, CDIs are positioned like large, global ISPs, but they do not use their network to transit traffic for other ISPs. This weakens the hierarchical Internet architecture and increases the flexibility of traffic. In the end, operational challenges faced by ISPs are multiplied, since traffic predictions become more challenging.

2.1.1 Traffic Engineering in an AS

The greatest challenge for an ISP is to keep its infrastructure operating efficiently. This is especially hard, since the ISP itself controls neither the behavior, nor the source nor destination of the majority of the traffic it carries. The destination of the traffic is determined by the end-users the ISP sells services to, while the source is usually operated by a CDI. The behavior is dictated through end-users requesting content, and by the operational choices of the CDI. ISPs today tackle the problem of network operation efficiency by performing Traffic engineering (TE). In its broadest sense, todays TE encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic [17]. Today, traffic engineering reduces to controlling and optimizing the routing function and to steering traffic on an Origin-Destination (OD) flow basis through the network in the most effective way.

Traffic Engineering encompasses multiple steps in order to be performed successfully. First, an ISP needs to record its traffic volume in terms of Origin-Destination flows. This means keeping traffic statistics on how much traffic flows from one node in the network to another. Once the OD flows have been successfully recorded, TE uses this information to simulate the network behavior with different IGP configurations. The goal of these simulations is to find an IGP configuration that spreads the network load as evenly as possible.

Figure 2.2 shows an example of how an Interior Gateway Protocol (IGP) configuration can be used to engineer traffic. The labeled circles represent routers, while the numbers in the squares represent the IGP-weight for the link. For ease of presentation, the weights for each link are set to the same value for both directions. An OD flow, which starts at one node and finishes at another, takes the path through the network that yields the smallest sum over all weights along the path. For example, in the starting configuration of the network (Figure 2.2 (left)) the flow IG does not take the direct path $I \rightarrow H \rightarrow G$ since, according to the IGP weights, a more effective path exists. In fact, the path $U \rightarrow H \rightarrow E \rightarrow D \rightarrow \ G$ has an accumulated weight of 4 instead of 5 (green path). All traffic at Router I destined for Router G takes this path. Similarly, all traffic that originates from B towards G follows the path $B \rightarrow E \rightarrow D \rightarrow G$ (blue path). Also, both paths share links, leading to a possible overload situation. In order to solve this problem, we choose to modify the link weight between the routers D and E. By increasing the weight from 1 to 5 (marked red in the right network), the blue as well as the green paths are shifted to the direct path. The change is shown in Figure 2.2 (right).

This simple diagram allows for illustrating multiple caveats that IGP based traffic engineering introduces. First, IGP-based traffic engineering affects traffic on an OD-flow basis only. This means that the path from one router to another can be changed, but the traffic on the OD flow cannot be split onto multiple paths. Secondly, the change of one weight can affect multiple OD-flows at the same time. Thus, the weights have to be changed very carefully. In the worst case, it might not be possible to fully separate some OD-flows due to the network layout.

The third caveat is not immediately obvious but needs to be considered very carefully when performing traffic engineering. While the link weights are usually known to all routers, they are propagated by messages that routers exchange. This propagation takes time, which can lead to short-term inconsistencies in the view of a network. We again use Figure 2.2 for illustrating this. When the link weight is changed, Routers D and E update their routing. This has an immediate effect on the traffic from B to G. With the update, the shortest path from router E to G is now $E \rightarrow H \rightarrow G$. In accordance, E configures its routing to send all traffic for G through H. However,

H has not converged at this point and still uses the old path. Thus, H still sends all traffic for G towards E. As long as H uses the outdated IGP weight information, all traffic for G that reaches either E or H is sent back and forth between the two routers. This forwarding, on the one hand, likely overloads the link. On the other hand, most traffic that is affected by this will be dropped due to its TTL running out.

The work of Francois et al. [49] shows that it is possible to gradually change IGP weights by sequentially ordering changes. Accordingly, routing loops like those in the example are avoided. However, these changes still require time during which the network can be in a transient state with overloaded links. Besides the challenges induced by optimizing the IGP, this approach also assumes that traffic is predictable and stable over time. By running simulations based on past traffic aggregates to engineer the network for the future, it is implicitly assumed that traffic patterns remain similar over a longer period of time.

With the emergence of CDIs, however, traffic has become volatile in terms of its origin. In fact, CDIs can shift their massive traffic amount in a matter of seconds from one server cluster to another. While this behavior is needed and propagated by CDIs, it is in stark contrast to the ISP's traffic engineering, which assumes traffic behavior to be stable for days, weeks or sometimes months.

2.1.2 Caching Traffic for AS Management

Since traditional link weight based traffic engineering is not able to dynamically handle the volatile traffic CDIs induce, ISPs have also tried to tackle this problem by caching content on proxies. In general, this concept entails that an end-user no longer connects directly to a content server, but is directed to a middle machine, called a proxy, instead. Proxies are scattered throughout the network of an ISP, usually close to end-users. Since hundreds, if not thousands of end-users use the same proxy, the content can be easily cached there. This outlines the idea simply: the proxy is able to store popular content and, once multiple users request it, can serve it directly without burdening the network by connecting to the CDI servers.

However, using proxies comes at a high operational and management cost. This is due to content having become very versatile and highly specific to interest groups. This means that content is not in general popular, but is specific to a culture, a social group, a language or a country. With more and more websites being driven by high volume user-generated content, such as Youtube, it becomes increasingly difficult for a proxy to cache the right content [8].

By installing proxies, ISPs also become part of the content delivery infrastructure. But as opposed to CDIs, ISPs are not paid for installing and operating this infrastructure. This leaves ISPs in a situation where they can neither steer the traffic in their network through configuration nor are they able to reduce the load on the

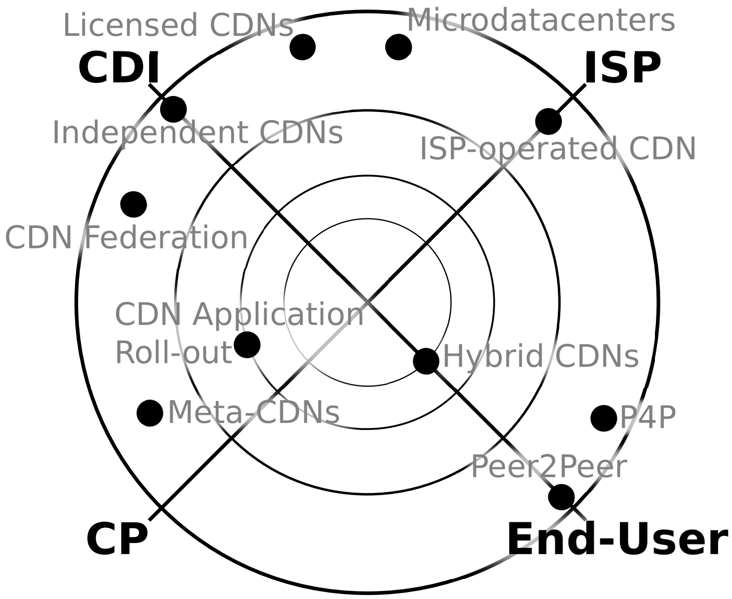


Figure 2.3: **Content Delivery Spectrum**

network by caching content at proxies. Furthermore, CDIs do not want ISPs to interfere with their traffic and assignment, the investment and operational burden for ISPs is high and the effectiveness gained by using proxies is limited.

2.2 Content Distribution Infrastructures (CDIs)

Next, we turn to the Content Distribution Infrastructures which dominate todays Internet traffic. In fact, they are responsible for more than half of the traffic in major ISPs [54, 107]. CDIs are overlays built on top of existing network infrastructures that aim to accelerate the delivery of content to end users. But not all CDIs are built upon the same philosophy and technology. For example, a CDI can be operated independently by deploying caches in different networks, by renting space in datacenters or by building its own datacenters. Other forms of CDIs are operated by ISPs, by content producers, or by self-organized end users. Figure 2.3 provides an overview of different CDI solutions. These are aligned by their architectures according to which parties are involved, and categorized by architectures into (a) traditional, which deliver the largest fraction of content today, and (b) emerging, which are newer but are beginning to have significant penetration in the content delivery market.

2.2.1 Independent Content Distribution

Independent CDIs are usually referred to as Content Delivery Networks (CDNs). They have a strong customer base of content producers and are responsible for delivering the content of their customers to end-users around the world. There are four main components to CDN architectures: a server deployment, a strategy for replicating content on servers, a mechanism for directing users to servers, and a system for collecting and processing server logs.

For server deployment, three main approaches exist [85]. The first is hosting in a single central location. This approach is used by small CDNs, One-Click Hosters, and applications running in public clouds. Centralized hosting takes advantage of (a) the economies of scale that a single location offers [16], (b) the flexibility that multihoming offers [55], and (c) the connectivity opportunities that IXPs offer [5]. The disadvantages of centralized hosting are the potential for a single point of failure, and the limited ability to ensure acceptable latency to users located in different networks around the world [87]. The second approach is to deploy in several large data centers. This approach leverages economies of scale while improving reliability. If datacenters are IXP members, they benefit from direct connectivity to a large number of networks [5]. This is the approach that CDNs such as Limelight, EdgeCast and BitGravity follow. Many cloud providers also use this approach, including Amazon CloudFront and Microsoft Azure. The third approach consists of a highly distributed deployment. This approach relies on a large number of servers scattered across numerous networks, offering high availability and replication of content. It can balance traffic across locations, react to flash crowds, and deliver improved latency. Akamai uses this approach.

CDNs typically follow a pull strategy [102] for content replication. When a requested object is not at the selected server, neighboring servers in the same cluster or region are asked. If the object is not available at neighboring servers, the origin or root server responsible for the object is contacted to retrieve the content. A requested object that is fetched from a remote server is saved locally and then delivered to the end user. To keep the copies of the object fresh, a TTL value is assigned to it. When the TTL value expires, the object is removed. For scalability reasons, any server of the CDN or within a region can respond to the request of an end user [129].

The independent CDI category also includes free CDNs such as Coral [50], which follow a similar architectural design. In these CDNs, server resources are offered by end-users or non-profit organizations.

2.2.1.1 ISP-operated CDIs

The potential for generating revenue from content delivery has motivated a number of ISPs to build and operate their own Content Distribution Infrastructures. For example, large ISPs such as AT&T and Verizon have built their own CDNs along the same general architectural principles as independent CDIs. However, due to the limitations arising from being restricted to one network, these CDNs are not deployed in a distributed fashion across multiple networks. To overcome this issue, the CDNi group at the IETF [100] is discussing how to interconnect these CDNs to boost their efficiency. The provider of the content can be third parties, applications and services offered by the ISP, or end users. Other ISPs with large footprints, such as Level3 and Telefonica [82, 84], have also built CDNs in order to efficiently transfer content across the globe.

2.2.1.2 Peer-to-Peer

The peer-to-peer (P2P) paradigm has been very successful in delivering content to end-users. BitTorrent [30] is the prime example, used mainly for file sharing. Despite the varying (perhaps declining) share of P2P traffic in different regions of the world [90], P2P traffic still constitutes a significant fraction of Internet traffic. P2P systems have been shown to scale application capacity well during flash crowds [139]. However, the strength of P2P systems, i. e., anybody can share anything over this technology, also turns out to be a weakness when it comes to content availability. In fact, mostly popular content is available on P2P networks, while older content disappears as users' interest in it declines. In the example of BitTorrent, this leads to torrents missing pieces, in which case a download can never be completed.

The positive aspects P2P systems derive from being distributed and decentralized is also their weakness when delivering small or time sensitive content. For example, bulk transfers take time, and it is not important whether the transaction takes a few more seconds, while websites and Internet based applications are very sensitive to page load delays. But since there is no central control of the content in P2P networks, it can take seconds, if not minutes until the content is distributed or a peer is found that can supply the content. Thus, P2P networks are not suited for delay sensitive content due to their unstructured, decentralized and unorganized nature.

2.2.2 Emerging Trends in CDI Architectures

Economics, especially cost reduction, is the key driving force behind alternative CDI architectures. The content delivery market has become highly competitive. While the demand for content delivery services is rising and the cost of bandwidth is decreasing, the profit margins of storage and processing [16] are dwindling, increasing the pressure on CDIs to reduce costs. At the same time, more parties are entering the market in new ways, looking to capture a slice of the revenue. However, today's traditional CDI deployments lack agility to combat these effects. Contracts for server deployments last for months or years and the available locations are typically limited to datacenters. The time required to install a new server today is in the order of weeks or months. Such timescales are too large to react to changes in demand. CDIs are therefore looking for new ways to expand or shrink their capacity, on demand, and especially at low cost.

2.2.2.1 Hybrid Content Distribution

In a hybrid CDI, end users download client software that assists with content distribution. As in P2P file-sharing systems, the content is broken into pieces and offered by both other users who have installed the client software as well as by the CDI's servers. The client software contacts dedicated CDI servers, called control plane servers, which schedule which parts of the content are to be downloaded from what peers. Criteria for selecting peers include AS-level proximity as well as the availability of the content. If no close peers are found, or if the download process from other peers significantly slows the content delivery process, the traditional CDI servers take over the content delivery job entirely. Akamai already offers NetSession [2], a hybrid CDI solution for delivering very large files such as software updates at lower cost to its customers. Xunlei [36], an application aggregator with high penetration in China, follows a similar paradigm. It is used to download various types of files including videos, executables, and even emails, and supports popular protocols such as HTTP, FTP, and RTSP. Xunlei maintains its own trackers and servers. A study of hybrid CDIs [65] showed that up to 80% of content delivery traffic can be outsourced from server-based delivery to end users, without significant degradation in total download time.

2.2.2.2 Licensed CDNs

Licensed CDNs have been proposed to combine the benefits of the large content-provider customer base of an independent CDI with the end user base of an ISP [13]. A licensed CDN is a partnership between an independent CDI and an ISP. The CDI licenses the content delivery software that runs on servers to the ISP while the ISP owns and operates the servers. The servers deliver content to the end users and report logging information back to the CDI. The revenue derived from content producers is then shared between the two parties. Thus, a CDI can expand its footprint deep inside an ISP network without investing in hardware, incurring lower operating costs. The ISP benefits from not having to invest in developing the software for a reliable and scalable content distribution. More importantly, a licensed CDN also alleviates the ISP's need to negotiate directly with content producers, which might be challenging, given an ISPs limited footprint.

2.2.2.3 Application-based CDIs

Recently, large application and content producers have rolled out their own CDIs, hosted in multiple large data centers. Some popular applications generate so much traffic that the content producers can better amortize delivery costs by doing content distribution themselves. Google is one such example. It has deployed a number of data centers and interconnected them with high speed backbone networks. Google connects its datacenters to a large number of ISPs via IXPs and also via private peering. Google has also launched the Google Global Cache (GGC) [56], which can be installed inside ISP networks. The GGC reduces the transit cost of small ISPs and those that are located in areas with limited connectivity, e. g., Africa. The GGC servers are given for free to the ISPs which install and maintain them. GGC also allows an ISP to advertise through BGP the prefixes of users that each GGC server should serve. As another example, Netflix, which is responsible for around 30% of the traffic in North America at certain times, is also rolling out its own CDI. The Netflix system is called Open Connect Network [99]. Netflix offers an interface where ISPs can advertise, via BGP, their preferences as to which subnets are served by which Open Connect Network servers.

2.2.2.4 Meta-CDIs

Today, content producers contract with multiple CDIs to deliver their content. To optimize for cost and performance [88], meta-CDIs act as brokers to help with CDI selection. These brokers collect performance metrics from a number of end users and try to estimate the best CDI, based on the server that a user is assigned. To this end, the brokers place a small file on a number of CDIs. Then they embed the request for this file in popular websites' source code, in the form of a javascript. When users visit these sites, they report back statistics based on the servers that each CDI assigned the users. The broker then recommends CDIs for a given source of demand taking also into consideration the cost of delivery. Cedexis is one of these brokers for web browsing. Another broker for video streaming is Conviva [38]. These brokers may compensate when a CDI does not assign a user to the optimal server (which a recent study [107] has shown sometimes occurs) by selecting a different CDI.

2.2.2.5 CDI Federations

To avoid the cost of providing a global footprint and perhaps to allow for a single negotiating unit with content providers, federations of CDIs have been proposed. In this architecture, smaller CDIs, perhaps operated by ISPs, join together to form a larger federated CDI. A CDI belonging to the federation can replicate content to a partner CDI in a location where it has no footprint. The CDI reduces its transit costs because it only has to send the object once to satisfy the demand for users in that location. Overall, cost may be reduced due to distance-based pricing [132]. The IETF CDNi working group [100] works on CDI federation.

2.3 Challenges in Content Delivery

Whether a Content Distributor Infrastructure (CDI) is using traditional or emerging solutions to deliver its content, some challenges remain constant throughout all of the settings. For example, it is hard for a CD to know where to do peering. Today, CDs solve challenges by peering with as many ISPs as possible, or deploying caches deep inside their networks. Also, for scalability reasons, most CDs make content available from all of their infrastructure locations [129]. This behavior of CDs is unlikely to change in the future due to multiple technical reasons, including administrative overhead, resilience, and management [85]. Also, with the globally deployed infrastructures and high inter connectivity to many networks, CDs are able to serve content from many different positions in a multitude of Networks and through many peering points. But this flexibility also introduces challenges to CDs, as stated in the following paragraphs.

Content Delivery Cost. CDIs 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 such as bandwidth or energy cost [55, 109]. While a CDI will try to assign end-users in such a way that the server can deliver reasonable performance, this does not always result in end-users being assigned to the server able to deliver the best performance. Moreover, the intense competition in the content delivery market has led to diminishing returns of delivering traffic to end-users.

End-user Mis-location. End-user mapping requests received by CDI DNS servers originate from the end-user's DNS resolver, not from the end-user itself. The assignment of end-users to servers is therefore based on the assumption that end-users are close to the used DNS resolvers. Recent studies have shown that in many cases, this assumption does not hold [6, 91]. As a result, the end-user is mis-located and the server assignment is not optimal. As a response, DNS extensions have been proposed to include the end-user IP information [32].

Network Bottlenecks. Despite their efforts to discover the paths between the end-users and their servers to predict performances [67], CDIs have limited information about actual network conditions. Tracking ever changing network conditions, i. e., through active measurements and end-user reports, incurs an extensive overhead for the CDI, without a guarantee of performance improvements for the end-user. Without sufficient information about network paths between the CDI servers and the end-user, an assignment performed by the CDI can lead to additional load on existing network bottlenecks, or create new ones.

End-user Performance. Applications delivered by CDIs often have requirements in terms of end-to-end delay [79]. Moreover, faster and more reliable content delivery results in higher revenues for e-commerce applications [102] as well as user engagement [38]. Despite the significant efforts of CDIs, end-user mis-location and the limited view of network bottlenecks are major obstacles to improve end-user performance.

3

Content Distribution Infrastructure (CDI) Deployment Study

One of the core observations this work builds on is that large CDIs are running distributed infrastructures to deliver content. Thus, the first question that arises is how CDIs deploy and operate their infrastructure as well as for how much traffic they are responsible. To highlight that the traffic is in fact delivered from multiple different servers at diverse network locations, we rely on passive network traces to identify popular services, active measurements to identify server location diversity and server logs from a CDN to confirm our findings.

3.1 CDI Operations from an ISP's Point of View

Today, most CDIs operate completely independent from ISPs. Thus, ISP have little information on CDI infrastructure, management and operational. At the same time, CDIs constantly increase their traffic and have become a major player on the Internet in terms of volume. Furthermore, CDNs employ a highly dynamic scheme to map users to their servers, inducing volatility to their traffic that is hard to manage with todays traffic engineering tools available to ISPs.

Our evaluation methodology relies on packet level traces from a large European ISP. We analyze them towards identifying CDI infrastructures and their behavior as seen by an ISP. Here, we find that CDIs rely on the domain Name System (DNS) for their operation. Thus, we focus our analysis on the DNS infrastructure in order to find the server deployment, mapping and operational behavior of CDIs. Based on

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

Table 3.1: **Summaries of anonymized traces from a European ISP**

these observations, we develop classification methods to detect CDI infrastructures and perform a first potential analysis on the impact of CDI operation when basic ISP knowledge is available.

3.1.1 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 [106] with dynamic protocol detection (DPD) [39].

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. 3.1 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.

With regards to the application mix, see Table 3.1, Maier et al. [90] 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 other recent Internet application mix studies [80, 90, 113, 118]. Figure 3.1 [117] 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.

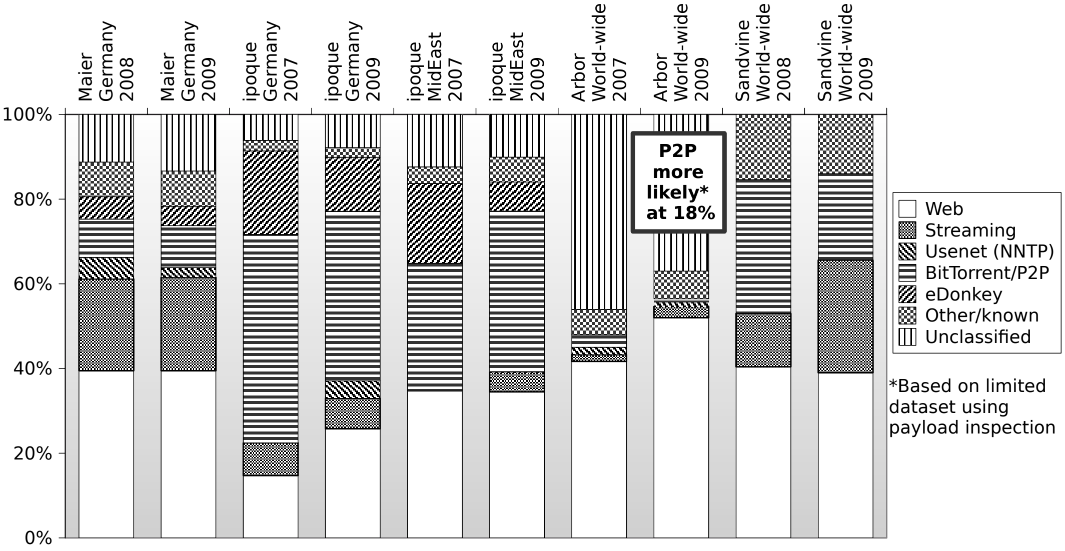


Figure 3.1: **Internet Application Mix (unified categories) across years and regions from multiple sources.[117]**

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 [90]. 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 [8, 90] and in MAR10, more than 50 % of the trace's total traffic is captured by these hosts.

3.1.2 CDI 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

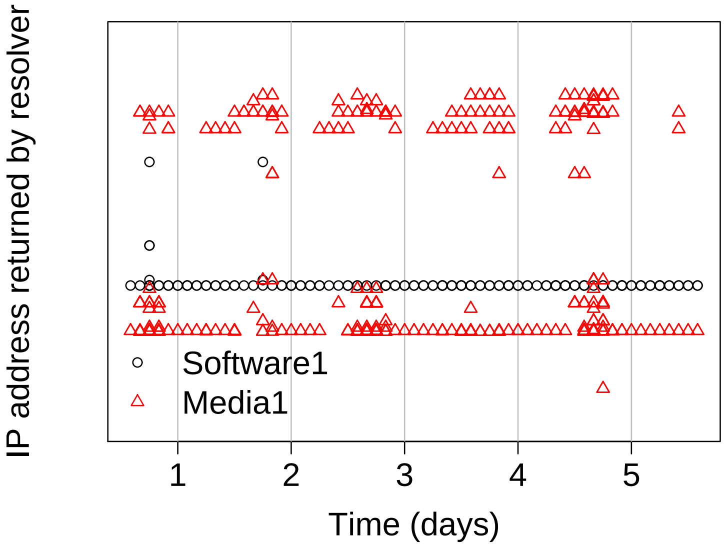


Figure 3.2: **DNS replies for two different sites hosted on a CDI, in two-hour bins**

usage patterns. We consider only the most heavily used resolver. Figure 3.2 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.

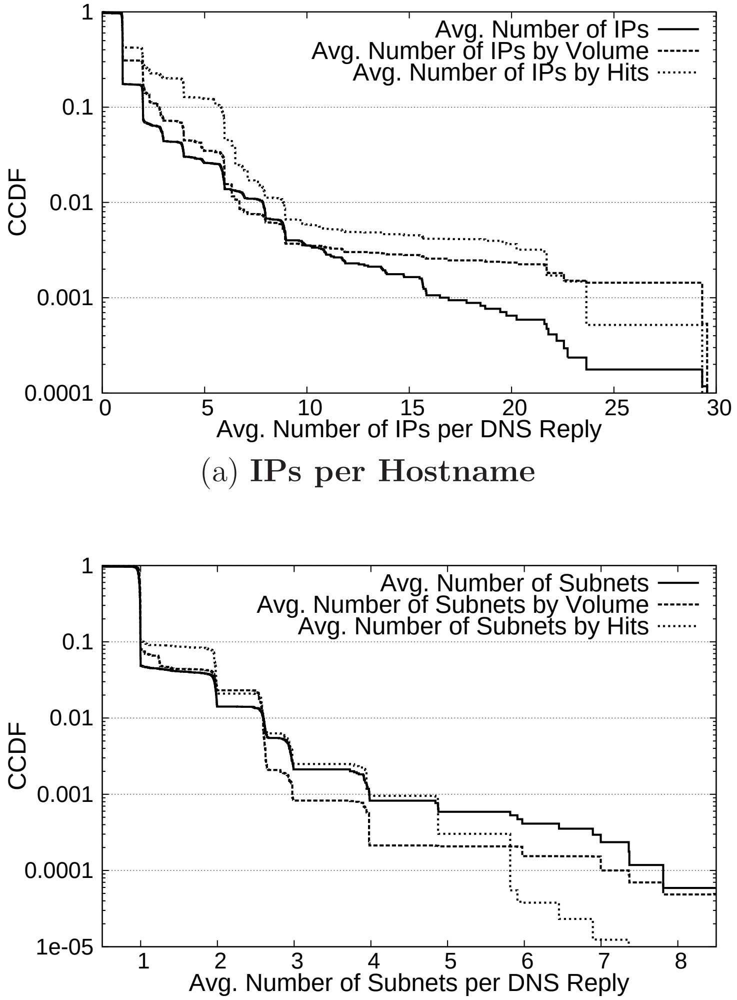
These results are promising, and show that individual sites do expose a certain degree of server diversity to their users. While our trace (HTTP-14d) includes the queried hostnames, it does not include the resolved IP address, as a HTTP request header contains the hostname but not the IP address of a 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. The querying of the hostnames is done sequentially, at a rate of roughly 13 queries/second to not overwhelm the DNS server and to avoid being blacklisted. The queries proceed through the set of hostnames acquired from HTTP-14d in a round robin fashion. However, if the previous DNS reply is still valid, i.e., if its DNS TTL has not yet expired this request is skipped and we proceed with the next hostname. In addition, we exclude those domains that consistently return NXDOMAIN or the default search site of the ISP. Together these are less than 3.5% of the hostnames. 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.

3.1.3 Server Location Diversity

Our analysis of hostnames and their assignment to servers in section 3.1.2 has shown that content is being served by multiple servers in different locations. In fact, many domains use the service of a *Content Distribution Infrastructure* (CDI), which can be seen during the name resolution progress: The original domain name is mapped to the domain of a CDI, which then answers requests on behalf of the requested domain name from one of its caches [127]. Almost all CDIs 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 CDIs is the independent CDN Akamai [66, 85, 127] as well as Google [79] including their YouTube service [22].

To better understand the DNS resolution process for hostnames hosted on CDI infrastructures, we refer to the machine requesting content as the `DNS client`. Along the same lines, we refer to 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, DNS resolvers act as caches to reduce the load on the authoritative DNS servers. Lastly, the authoritative DNS server, henceforth referred as `DNS server`, which is usually run by the CDI, is responsible to select a CDI server that handles the content request of the client. Note, that at the point where the request reaches this server, no content has been transferred to the client. So far, only the mapping of the hostname to an IP address has taken place. Once the authoritative DNS server of the CDI has chosen the CDI servers that can deliver the content, it generates a reply and sends it back to the DNS resolver. The DNS resolver caches the reply and hands it back to the DNS client.

While this scheme follows the standard DNS resolution process for the most part, it also introduces an additional step. For CDIs, a simple database lookup to resolve the hostname is not enough. Since CDIs have hundreds, if not thousands of servers at their disposal, they need to balance the requests between these. Therefore, the DNS server has to choose which and how many IP addresses it returns. Furthermore, a static resolution process of hostname to IP address is not agile enough. Therefore, additional information becomes part of the server selection process. Furthermore, additional information may be used to help select a close-by server. Examples include the hostname that was being requested, IP-Geolocation databases [122] to localize the region of the DNS resolver, BGP data to identify the ISP and a topology map derived via traceroutes, or any combination of these. Finally, a DNS server has,



(a) **IPs per Hostname**

(b) **Subnets per DNS Hostname**

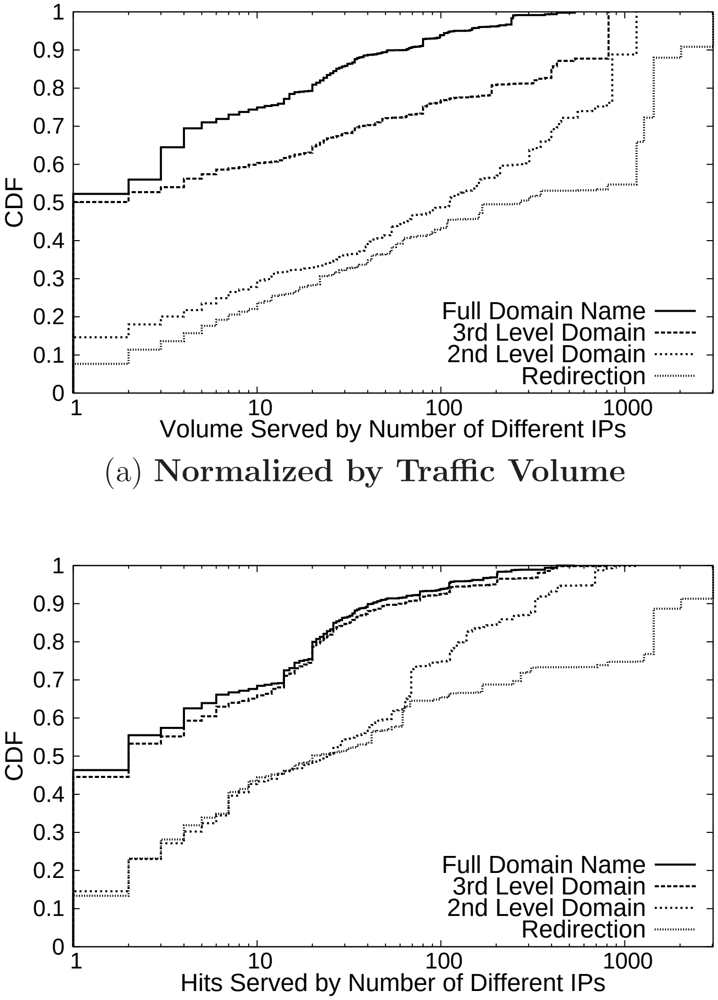
Figure 3.3: **Average Number of IPs and Subnets for DNS Hostnames**

in principle, two methods for load balancing across the multitude of servers it is choosing from:

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



(a) **Normalized by Traffic Volume**

(b) **Normalized by Requests**

Figure 3.4: **CDF of unique IPs per Hostname and Aggregation**

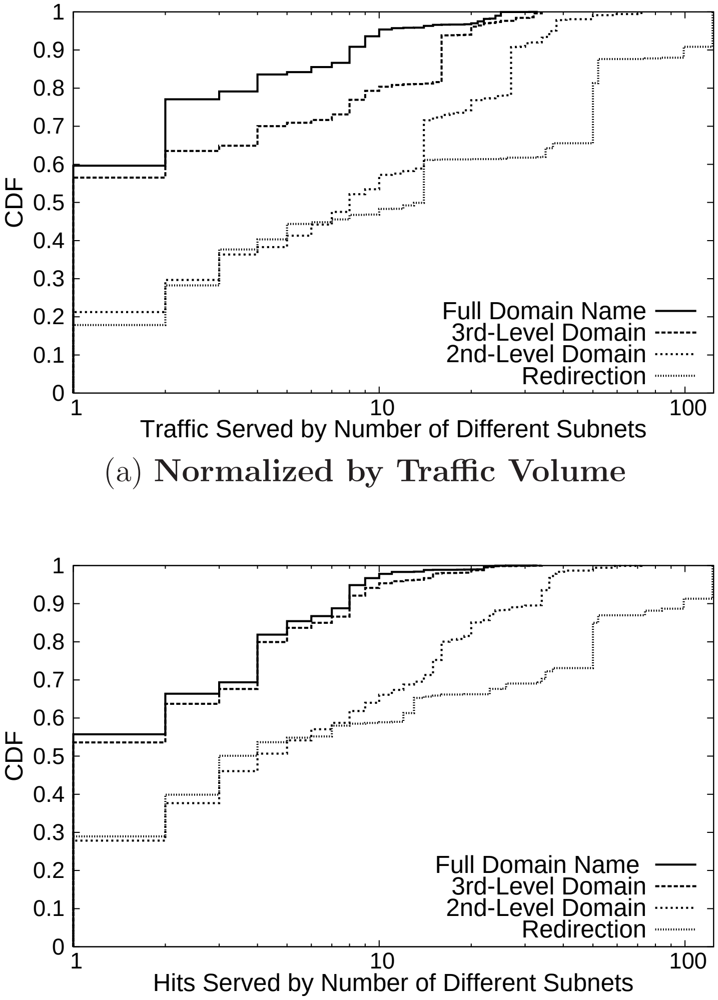
BGP information obtained from within the ISP. This allows for detailed information about the different locations within the ISP, while giving an aggregated view of subnets reachable via peering links. Another issue stems from the fact that the IP address returned by the CDI is influenced by the IP address of the ISP DNS resolver [6, 105, 127]. Due to this, we used the DNS resolver of the ISP of our vantage point as well as external DNS resolvers (see section 3.1.4). 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 CDI for this specific ISP.

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

a) Prevalence of MultQuery We start our analysis by checking the prevalence of the first form of DNS based load balancing, MultQuery. The CCDFs show the average number of IP addresses (Figure 3.3a) and subnets (Figure 3.3a) 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 quantile is 2. 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. Next, the diversity in subnets is being evaluated. Here, we find 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 the granularity of 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.

b) Prevalence of CrossQuery Next, we check how prevalent CrossQuery, the second form of DNS based load balancing is. Since CrossQuery potentially returns different IP addresses for repeated queries, its contribution to server diversity can only be studied by aggregating DNS replies over time. The lines labeled `Full Domain Name` in Figures 3.4 and 3.5 capture this case. In contrast to only evaluating a single reply, we find that in this case more than 50 % of the volume or requests can be served by more than one IP address. Similarly, there is choice of at least two subnets over 40 % of the time across both metrics, see Figure 3.5. This indicates that there is significant potential for the ISP to bias the location preference of the CDI.

c) Subdomain Aggregation Since some CDIs 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 3.4 and 3.5 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 for large CDIs, 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 find that at least two major independent CDNs, a streaming provider and a One-Click Hoster serve requested content from servers that match in their second level domain. However, for smaller infrastructures, this approach can over-estimate the diversity of servers for a given domain.



(a) **Normalized by Traffic Volume**

(b) **Normalized by Requests**

Figure 3.5: **CDF of unique Subnets per Hostname and Aggregation**

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 and by volume. The number of returned subnets further increases when accumulating to the second-level domain of DNS resolver answer. Studying our traces in detail, we find this is due to the substantial traffic volume and number of requests served by CDIs, some of which are highly distributed within ISPs or located in multi-homed datacenters or peer-exchange points.

d) Infrastructure Redirection Aggregation To counter the over estimation introduced by the subdomain aggregation, we take a closer look at the DNS replies [95]. Here, we find that some CDIs use CNAME chains to map queried hostnames to an

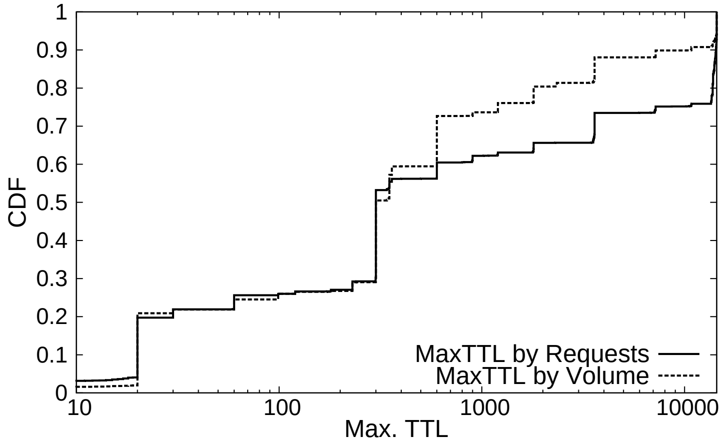


Figure 3.6: **CDF of DNS TTL by traffic volume and by number of requests**

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 [129]. On the down side, it is possible that this scheme of aggregation reduces the effectiveness of the CDI's caching strategy. This aggregation is called `Redirection` in Figures 3.4 and 3.5.

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. Furthermore, if we consider aggregation periods in the order of tens of minutes, the effects on the server diversity are only minor. The reason that most of the diversity is observable even over these short aggregation time periods, is that the typical TTL, see Figure 3.6, 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.

3.1.4 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 CDI is biased by the querying DNS resolver, we repeat the experiment from Section 3.1.2 using two other DNS resolvers. In particular, we pick the next most

	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 %

Table 3.2: **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.**

popular DNS resolvers found in our traces: GoogleDNS and OpenDNS. Both are anycasted third-party resolvers with a global footprint. 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 CDIs select the supplied caches based on the source IP address of the querying DNS resolver. Since the CDIs are no longer able to map the request to the AS it originates from, but rather to the AS the DNS resolver belongs to, the server selection of the CDI cannot optimize for the location of the DNS client even on an AS level. In many cases, the CDI has to fall back to generic positions that are near the third-party resolver, but not related close to the user.

3.1.5 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 3.1.3 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 3.2 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, i. e., GoogleDNS or OpenDNS. 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 CDIs may always return IP addresses outside the ISP, despite the fact that many of its servers are deployed within the ISP. The reason behind this is that the CDIs cannot map the DNS resolver to the AS anymore, and thus are unaware of the origin of the request. This explains the substantial difference and highlights on the one hand the effectiveness of the CDI 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 [32].

However, one can ask if the CDI utilizes the full potential of traffic localization on an AS level. 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 3.2 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 collaboration between CDIs and ISPs. While the increase is noticeable 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 infrastructures can be localized the most.

3.1.6 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 3.1.3 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. 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.

Log type	Entries
CDN Connection Log	ca. 62 Million valid connections March 7th 2011 - March 21st 2011
ISP Backbone	650+ Nodes
ISP Routing	ca. 5 Million entries

Table 3.3: **Datasets: CDN and tier-1 ISP**

Server	Inside	Outside
Users Inside	45.6%	16.8%
Users Outside	37.6%	N/A

Table 3.4: **Fraction of CDN traffic inside ISP**

3.2 Current Service Deployment of an Independent CDN in an ISP

To further augment the analysis of CDI operations, we turn our attention to the analysis to one CDI and how its operation works inside one ISP. This CDI is a the largest independent CDN as of today. We use operational traces from the CDN and a tier-1 ISP (see Table 3.3 for details about the datasets.) to demonstrate the current service deployment and where a collaboration of the CDN and the ISP can benefit the parties.

Dataset–CDN: The dataset from the CDN covers a two-week period. It records the details of TCP connections between the CDN's servers and end-user machines, which are randomly sampled and reported to the log. Furthermore, the log has been reduced from the global footprint of the CDN to only include entries related to the tier-1 ISP. This means that either the client or the server is using an IP address that belongs to the address space of the ISP. When analyzing the CDN infrastructure, we find that the CDN operates on a global scale in more than 100 Networks. Finally, we confirm that the CDN also operates clusters, i.e., server racks, in a subset of the tier-1 ISPs locations. For each reported connection, the log contains the time at which it was recorded, the address of the server inside the cluster that handled the connection, the cluster that the server belongs to, the anonymized address of the client that connected to the server, and various connection statistics such as bytes sent/received, duration, packet count, RTT, etc.

Dataset–ISP: The second dataset contains detailed network information about the tier-1 ISP. It includes the detailed backbone topology, with interfaces and link annotations such as routing weights, as well as nominal bandwidth and delay. Furthermore, it also contains the full internal routing table which includes all subnets propagated inside the ISP either from internal routers or learned from peerings.

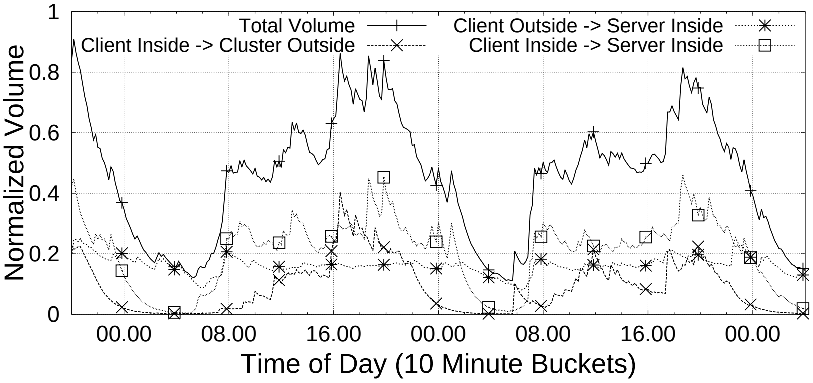


Figure 3.7: **Moving average ($\alpha = 0.85$) of normalized CDN traffic by origin/destination position across two days**

Thus, the full backbone topology as well a the entire internal routing is known for the analysis. The ISP operates more than 650 routers and 30 peering points all over the world. Finally, due to the included infrastructure information, the dataset enables reliable geographical mapping of the IP address space to city level resolution. Due to the layout of the ISP, any finer grained information is not available, since all subscriptions are running through central points-of-presence at these cities.

3.2.1 Current Operation

Table 3.4 shows the fraction of CDN traffic broken down by its origin and destination. The total traffic observed includes requests from end-users subscribed to the ISP to CDN clusters inside and outside the ISP, as well as requests from end-users outside the ISP that are served by CDN clusters inside the ISP. Since the dataset was restricted to only consider CDN traffic that relates to the ISP, no traffic is included when both, the end-user and the server are outside the ISP.

For this specific CDN, we find that 45.6% of its traffic is served by clusters within the ISP to end-users in the ISP. Comparing this to the earlier, more general analysis (see 3.1.5) of the entire traffic, it shows that this particular CDN is already keeping twice as much of its traffic local than the average. However, 16.8% of the total traffic is served to end-users in the ISP by clusters outside the ISP. This could be due to the fact that the CDN servers inside the ISP reached their limit and cannot serve any more traffic, forcing the CDN to map the incoming requests outside the ISP. But the fact that 37.6% of the total traffic served by clusters inside the ISP goes to end-users outside the ISP voids this assumption. In fact, some of the clusters inside the ISP serve external requests while requests from within the ISP are routed to servers outside the ISP. Indeed, closer inspection shows that some clusters are

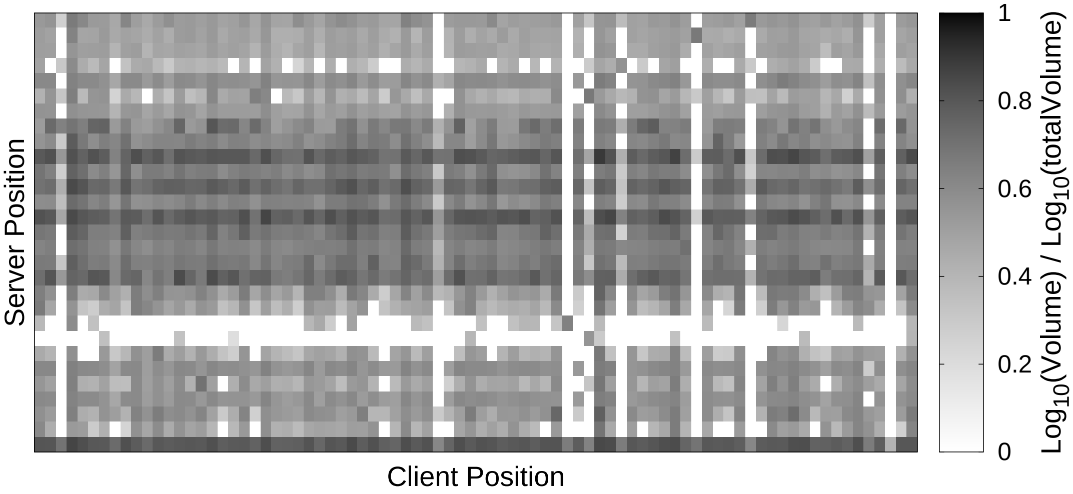


Figure 3.8: **Traffic heatmap—cluster vs. client position**

mainly used by the CDN to serve requests from ISPs in which the CDN has no physical presence nearby.

However, Table 3.4 is an aggregate over the entire time of the dataset. To get a better look at the dynamics of the CDN traffic, Figure 3.7 plots the percentages over a period of 60 hours. In order to counteract the effects of confining a data entry to one bin when it might span multiple as well as to smooth the traffic curve, the moving average function has been applied to the plot. During peak times, the percentage of the total volume from CDN servers external to the ISP to clients inside the ISP can reach up to 40%. This indicates that there is a significant fraction of volume from end-users within the ISP that is served from servers outside the ISP even though there is in principle capacity available within the ISP. At the same time, this capacity is being used by end-users from outside the ISP.

3.2.2 Mapping End-Users to Server Location

Next, we take a closer look at the 62.4% of CDN traffic to clients within the ISP. For this, we need to introduce the notion of a location. CDN servers deployed in a datacenter or co-location center are connected through an edge router redundantly to several co-located border routers. For the sake of the analysis in this section, these border routers constitute the location of the CDN servers. Similarly, end-users are connected to an edge router via an access network and this edge router is connected to several border routers. The location of the end-user, then, is taken to be that of the border routers. Currently, the CDN operates more than 35 clusters within the ISP, across more than 25 locations among the more than 100 major PoPs. (Note, when edge routers are considered locations, there are approximately an order of magnitude more locations.) Presumably, most requests from end-users at an ISP location with a co-located CDN cluster should be served locally. Moreover, locations where the CDN does not operate a cluster should be served from a nearby location. There will of course be exceptions. For example, for load balancing purposes, the CDN may have to map end-users to a cluster other than the nearest cluster. Or perhaps the CDN's active measurements indicate that better performance can be gained by mapping to a different cluster.

To check this presumption, Figure 3.8 plots a heatmap of the relative volume (logarithmic) of the mapping of ISP client locations to CDN locations. There are very few holes in this matrix, meaning that end-users at a given location are mapped to a variety of different locations over the 14-day period that we analyzed. Also, when looking at individual client positions, there are only a few that clearly have a preference in terms of what servers they are using. Most of the clients are being mapped to almost any server location. Also, there are four server positions which are notably more used than any other. The top three are large datacenters inside the ISP. However, the bottom row of the matrix shows the traffic that is served from CDN clusters outside the ISP to end-users inside the ISP. Finally, we observe that all end-user locations are mapped, at some point, to CDN servers outside the ISP network.

3.2.3 Location of CDN Servers

We have seen that the CDN maps end-users from one location to many different cluster locations over the course of two weeks. Several explanations can be envisioned, including: (a) the capacities at the CDN server locations do not match the volume of demand from the end-user locations, and (b) the mapping is suboptimal due to insufficient knowledge by the CDN of the network structure.

Figure 3.9 shows the fraction of traffic delivered by the CDN to each location within the ISP. Furthermore, for each location it is marked if there is a cluster of the CDN present (black) or if one is missing (white). Overall, we observe significant variations in terms of demand to the CDN. Some locations originate high demand while others have limited demand, if any. Closer inspection reveals that some of the locations with high demand do not have CDN servers nearby. Moreover, looking at the demands across time reveals even more mismatches between demand and CDN servers locations. A changing demand over time is explanation for such mismatches. Demand might have moved from one location to another one. However, changing the location of CDN deployments is cumbersome. In addition, at the time of the CDN deployment, some of the ISP locations with high demand might be unable to accommodate CDN clusters.

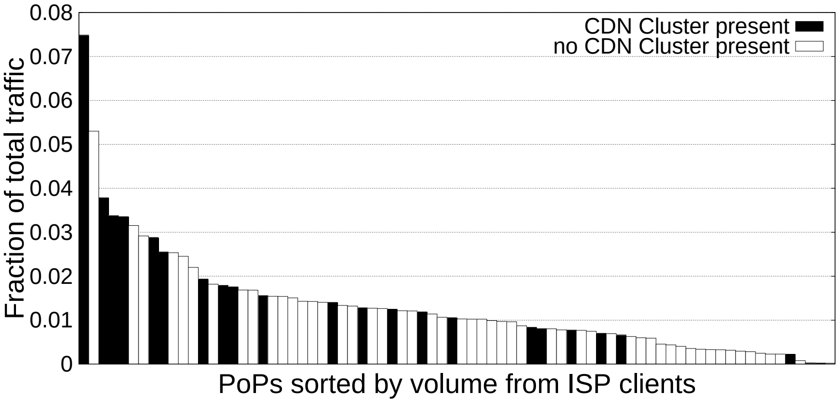


Figure 3.9: **Traffic demand by ISP network position**

3.3 Summary

Today, pure CDI architectures have ceased to exist. The classic CDI operation has been split into several sub-classes, all with their own distinct mode of operation, e. g., deep cache deployment, datacenters, peering or ISP owned CDIs. Additionally, there are new technologies pushing into the market, like hybrid CDI-P2P, CDI federations or Meta CDIs. However, all of these CDI solutions have common challenges: They struggle with the detection of the network topology and its current state.

To this end, we performed an analysis of the general HTTP behavior in an ISP. Through this it became visible that CDIs have multiple ways of load balancing users across their infrastructure. In most cases, this is done by dynamically adjusting the DNS replies to end-users. First, we find that the usage of third party resolvers significantly reduces the ability of CDIs to map users to the correct network. Next, we turn our attention to the amount of traffic that stays local to the ISP, and compare this to the traffic that could, in principle, stay local if the CDI was collaborating with the ISP to map the users. The potential in this case is tremendous. It shows that almost half of the analyzed traffic could stay local to the ISP, which is almost double to what is being delivered locally today.

We investigate this potential further by analyzing the behavior of a large independent CDN inside the same ISP. Here, we find that this particular CDN already does a good job at keeping the traffic local, but does not manage fully. In fact, more than a quarter of the traffic that originates from the end-users of the ISP are still being served by users outside of the ISP. Finally, we show that the positions of the CDN servers inside the ISP is not optimal in terms of the demand distribution.

Thus, we conclude that by collaborating with ISP, a CDI can significantly improve its operation. Furthermore, we find that ISPs can not only help with giving CDI insight into the network, but can also offer additional information towards improving the long term deployment by suggesting where to place servers most efficiently.

4 Using IP-Geolocation for CDI Operations

One of the biggest challenges CDIs face is mapping end-users to a nearby server. There are several methods of making the match: AS-Level, measurements such as pings and traceroutes, IP-Geolocation and a mixture of all of the above. This section explores the potential of using IP-Geolocation to map servers. To this end, it first uses a dataset of user submitted GPS locations from a large CDI in order to see how many IPs can be geolocated, and to what degree. This dataset is then augmented with an IP-Geolocation database, and finally, active measurements are used to further increase confidence in the results. In the second part, the focus turns to comparing multiple, already existing geolocation databases. Finally, the geolocation databases are compared to the ground truth information obtained from the tier-1 ISP (see section 3.2).

4.1 IP-Geolocation

IP-Geolocation attempts to find the geographical position of an IP address on the globe. By extension, it assumes that the user associated with the IP address is also close to IP's location. Thus, the inference is that by locating an IP address, the location of the user associated with that IP address is found. Today, this information is used for several purposes, such as localized online advertisement, content restriction based on country or continent or in the selection of a nearby server in CDI operation.

In the simplest cases, IP-Geolocation is achieved through static information from registries or active network measurements. When location information from static sources is used, it usually contains an overview on who owns the network associated with a specific IP address and where the company is located, but it does not offer information about where individual IP addresses are assigned. An example for this is a large, globally deployed ISP. It holds a significant portion of the IP address space; all registered to the ISP at the location of its headquarters. However, the IP's actual assignment to devices is an entirely different issue. This can lead to stark contrasts between the supposed and the actual Position of an IP address.

Another common method for achieving IP-Geolocation is through active network probing. The simplest approach is to run round trip measurements between a well known location and the target IP. The time it takes for the answer to return to the source limits the maximum distance between between the two points. However, these measurements are tedious. Firstly, the IP address space is huge and probing each IP address is cumbersome, even with multiple probes and aggregation to more coarse subnets [64]. Furthermore, active network measurements suffer from multiple inaccuracies, such as unresponsive IPs, packet loss, queuing delay, and the fact that the network distance is not equal to the geographical distance. Thus, determining the location of an IP address through measurement offers an upper boundary to how far an IP address is away from a specific point. By probing from different geographical positions, the area of an IP address can be confined, but due to the inaccuracies of probing, a reliable position is hard to obtain.

A third option for obtaining IP-Geolocation information is by using specialized databases. This approach offers the advantage that databases do not need large scale measurements and a distributed network infrastructure. Also, querying a database is orders of magnitudes faster than running active network measurements. However, while IP-Geolocation databases are easy to use, it is unknown how they are built and maintained. Thus, it is possible that they suffer from the same inaccuracies as active measurements, as they might be built the same way.

Finally, the location of a device can be obtained by the device itself. Modern smart phones, tablets and personal computers are either already equipped with location devices or these features are easily obtained. Through this, a device can, in theory, reliably supply its own location whenever a location service is needed. The most common way for a device is to acquire its position is through assisted-GPS (aGPS). How exactly this works differs from device to device, but the general idea is the same for all. If GPS is available, this position is used. If GPS is not available, which is the case whenever no direct signal from the sky can be received, the device falls back to other methods of inference. This can include an IP-Geolocation database lookup based on its own IP address or an inferred position based on nearby Wifi networks. Unfortunately, most devices do not reveal to the service through which technique they obtain their location.

IP address (server)
IP address (client)
Timestamp (epoch)
Latitude
Longitude
User ID (hash)

Table 4.1: **Server logs fields**

Start date	01.01.2010
End date	14.02.2012
log entries	ca. 1.1 Billion
Unique IPs	ca. 16 Million
Unique IDs	ca. 10 Million

Table 4.2: **Dataset Statistics**

4.2 Dataset: GPS Locations from a Media Broadcast Service (MBS)

In order to evaluate the applicability of IP-Geolocation for reliable end-user position inference, we first turn our attention to a study regarding user submitted locations. Here, we pose the question of how reliable a location submitted by a device is, in terms of selecting a close-by CDI server. This reverses one of the basic principles of IP-Geolocation: While geolocation knows the IP address and infers the location, this datasets knows the position and matches it to the IP address of the device.

We base the evaluation on a dataset that consists of HTTP connections from users to a Media Broadcast Service (MBS) run by an independent CDN. It is based on connection logs on the server side of the infrastructure. Specifically, each subscription an end-user issues to the MBS carries additional information. This includes the basic connection statistics, such as IP addresses and ports, as well as the supposed device location, media stream, device model, OS and version. One of the possible options in the URI that can be submitted by the user is a GPS position. But not all users supply the necessary information for this study. The minimum set of information needed for this study is shown in Table 4.1. Therefore, we filter the dataset to remove all connections that do not meet these requirements. Table 4.2 shows the general statistics of the remaining dataset after running the filtering process.

The dataset only entails the initial connection establishment information. There are no updates if the user moves while being subscribed to the MBS as long as the underlying TCP connection is not interrupted. A new log entry is only recorded if the user decides to subscribe to another media channel, gets disconnected or has a short network outage that requires re-establishment of the TCP connection.

4.2.1 Dataset Overview

With about 1.1 billion records having reported GPS locations over more than two years, the first question to arise is where these locations are and what the bias towards specific regions, i. e., countries, languages, cultures, etc., is. Figure 4.1 shows the dataset on a world map 4.1.

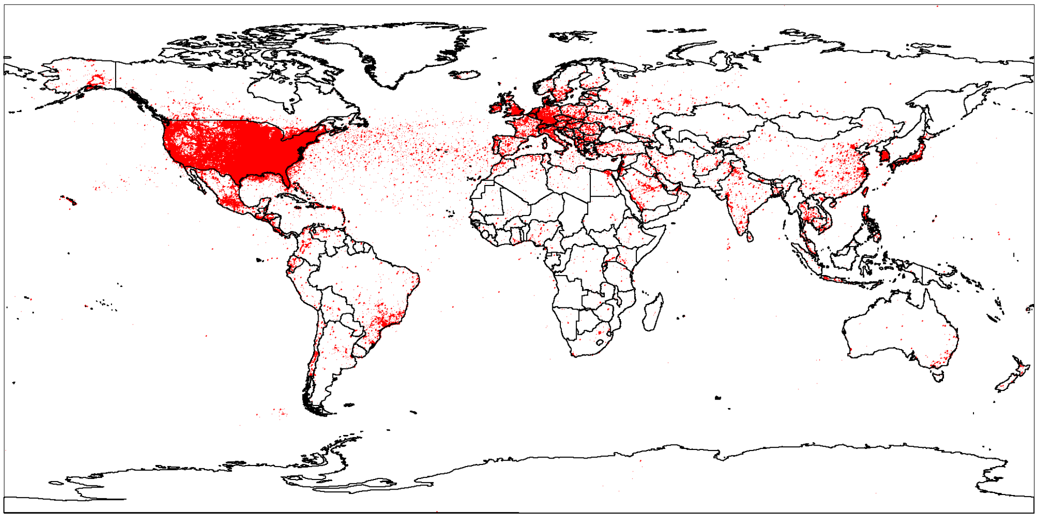


Figure 4.1: **All reported positions around the world. Most reported locations are from the US, central Europe, South Korea, Japan and the Atlantic Ocean**

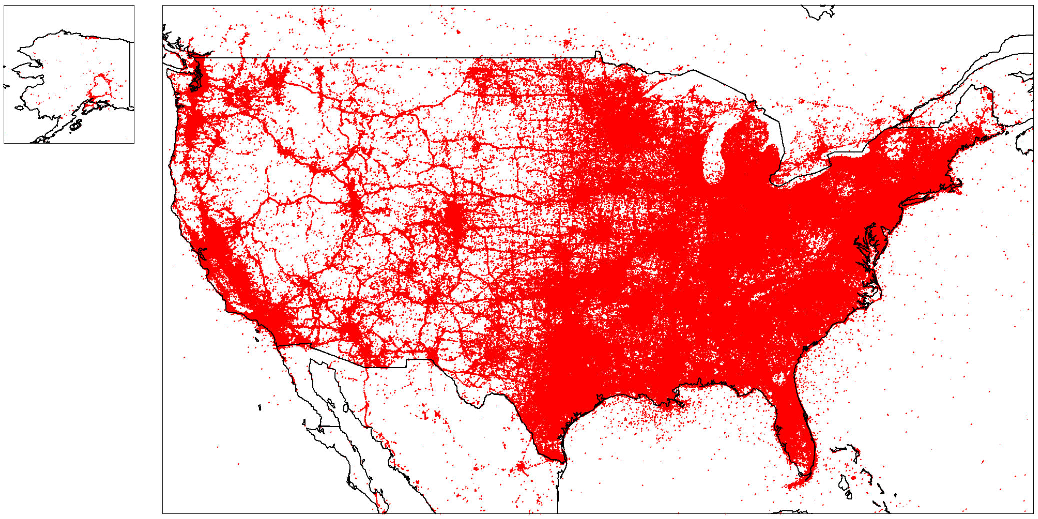


Figure 4.2: **Zoom on locations in the US with major traffic routes showing in sparsely populated areas**

Country	Inside	Country	Inside
USA	96.64%	Germany	0.08%
None	2.07%	Canada	0.07%
Japan	0.16%	Brazil	0.06%
Korea	0.12%	Saudi Arabia	0.05 %
Mexico	0.1%	Singapore	0.03%

Table 4.3: **Top 10 Countries with # of positions**

The most prominent feature about the map is that there are hot-spots where users are using the MBS. These locations include the US, Europe, South Korea and Japan. However, the MBS is obviously US-centric, placing more than 96% of the reported locations there. Curiously, the second largest contributor is not a country, see Table 4.3. In fact, more than 2% of the locations are reported from outside any country borders. Most of these locations are in the Atlantic ocean between Europe and the US. Also, a trail from US to the island of Hawaii is visible. Possible explanations for them are devices that are on ships or airplanes with Internet access. Also it is possible that some devices "spoof" their geolocation [128].

Turning back to the countries, it is shown that the majority are reported from inside the US. Figure 4.2 shows a zoom on the US. It comes as no surprise that the distribution reflects the population. However, an encouraging detail is that not only the coverage of densely populated areas are given, but also major highways are shown on the map. This can be be most prominently seen in Alaska, where the two major highways between the only two larger cities are clearly visible.

4.2.2 IP-to-GPS Correlation

Now, let us look at the correlation of IP addresses to the reported GPS locations. For this, we turn to the overview on how many locations are reported by each IP. Figure 4.3 shows the CDF of locations associated with the same IP address. Our first observation is that there is a significant amount of IP addresses that have very few samples, while a few have more than a million reported locations. We find that a substantial amount of IPs (35%) have more than 10 reported locations while 20% have more than 25 locations associated with them. Note that the Figure has been cut off at 10.000 IP addresses for better readability.

4.2.2.1 Unique Locations from End-Users

It is curious that the number of unique locations per IP address is significantly lower than the total number of locations. This means that devices are reporting the exact same locations multiple times. However, more than 99% of the locations are reported

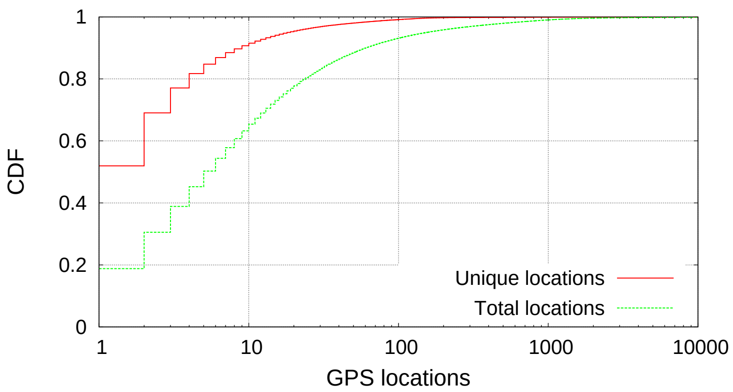


Figure 4.3: **CDF of number of Positions reported by Users**

to the 6 decimal digit, what translates to less than a meter in terms of geographical distance. In contrast, the current GPS specifications [97] state that the maximum accuracy in 95% of the measurement is at most 7.8 meters. Therefore, measured GPS locations should jitter due to measurement inaccuracy in the last digits.

There are several possible reasons for a device reporting the exact same location multiple times. First, mobile devices sample their location in intervals and not on a per request basis. If the user re-connects to the MBS multiple times without the device updating its position, it is possible to see the exact same location multiple times. This means that if multiple connections to the Media service are made while the GPS location is not updated, the exact same position is reported multiple times. This hypothesis can be checked by analyzing the interval with which a device reconnect to the MBS.

Figure 4.4 shows the time between the first and the last record for an IP address. We observe that more than 50% of all IP addresses show a duration of at least one hour between the first and last occurrence. When restricting the analysis to only take IP addresses into account that report one unique location, the picture changes. First, we observe that 38.7% of the IPs only have one location. These are all IPs that have a duration of exactly 0. For the rest we observe that almost 40% of have a duration of more than 0.1 hours, or 6 minutes. Thus, the assumption that the device does not update its location before the re-establishment of the connection does not hold. Another reason for the exact match of the GPS positions is aGPS inference of the location. The location inference through aGPS severely reduces the dataset. While over 80% of the IPs report more than one location, only about 50% report more than one unique location.

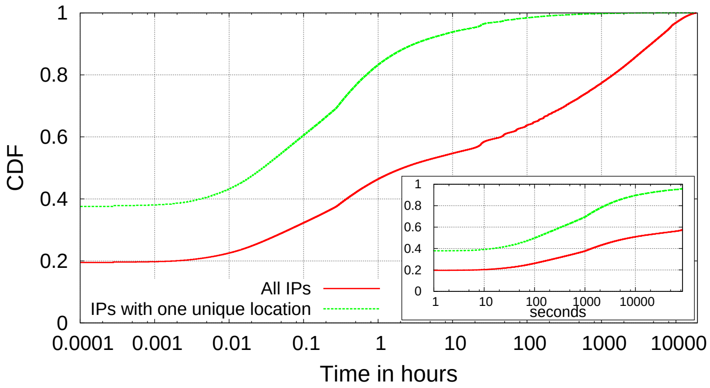


Figure 4.4: **CDF of the time between the first and last occurance of IP addresses**

4.2.2.2 Spread of Locations

Next, we turn to the spread of locations. By determining the pairwise distances between any two locations associated with the same IP address, it is possible to confine the area of an IP. But calculating the pairwise distances turns the n known locations into a set of $\frac{n-1}{2}$ distances for each IP address. Figure 4.5 captures the trend of the distance sets through the CDFs of several quantiles over all IPs. For example, the CDF labeled 5^{th} is the CDF of the 5^{th} quantile across all distance sets. Along the same line, the CDFs labels 25^{th} , 75^{th} and 95^{th} are the corresponding quantiles. The CDF label min refers to the minimal distance between any two coordinates, while max refers to the maximum value in the set. Finally, the $median$ is self-explanatory and the avg is the arithmetic mean over all distances in the set for each IP. The graph has been cut off at 100 meters, since anything with a distance of 100 meters is considered to be sufficiently accurate . On the other end of the scale, it is impossible to have distances larger than 21.000km due to the size of the earth. By evaluating at the maximum distance between any two locations for the same IP address, the area in which an IP address is located can be confined. Naturally, the smaller the maximum distance, the more accurate the location is.

Turning to Figure 4.5, we notice that about 50% of all IPs have a maximum location spread of less than 0.1km. Without the cut off at 0.1, it shows that 51.9% of all IPs report a distance of exactly 0. However, this result is to be expected. 18.8% of all IPs only have one location, what defaults to a maximum distance of 0. Furthermore, 33.1% of all IPs report the exact same location multiple times. This again defaults to a maximum distance of 0. Adding these two classes of IP address together reveals that 51.9% of the IP addresses must have a maximum distance of exactly 0. This corresponds almost exactly to the number of IPs that have a distance smaller than 0.1km. Thus, there are only a very limited number of IPs that have a maximum distance greater than 0 and less than 0.1km.

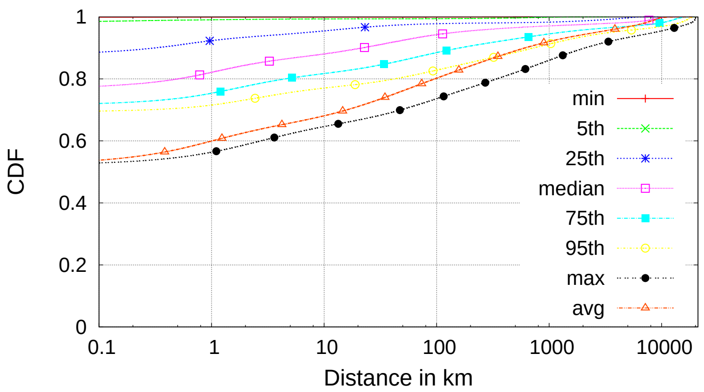


Figure 4.5: **Maximum Distance between positions reported by the same user**

Turning to the remaining 48.1% of the IPs that see more than one unique position, we observe that the distances between grow rapidly. 64% of all IP have a maximum distance of 10km, while the maximum of 25km applies to 67% of the IPs. Subtracting the known 0-distance entries from this shows that a 10km confinement is possible 12.1% of the IPs, while 25km increases the number to 15.1%.

Turning to the high distance area, we find 6.3% of IPs show a maximum spread of locations to be more than 5000km. This, for the US, is already the full country, while in Europe this distance spans several countries and most of the continent. This observation illustrates that the correlation between a device-derived location to an IP address cannot be made lightly. Another interesting observation is that the maximum and the 95 th percentile are far apart. This indicates that, while some locations are far apart, most of them seem to be significantly closer than the maximum distance. Also, almost all IP addresses have a very small minimum distance, indicating that it is common for devices to send close by locations.

4.2.2.3 Correlating IPs with IDs

There are multiple reasons for the large distances between locations associated with the same IP. The most likely is the deployment of proxies and gateways. Especially in cellular networks, middle machines are used to aggregate users [130]. To check if users are being aggregated by proxies and gateways, we turn to the user IDs associated with each log entry. Since the user IDs are unique to a device, this approach allows for an overview on how many devices are associated with any IP address. A large number indicates aggregation of users by a middle machine, while a small number is likely to be used by home subscription or, in the case of exactly one user-ID, a device that owns the IP address itself.

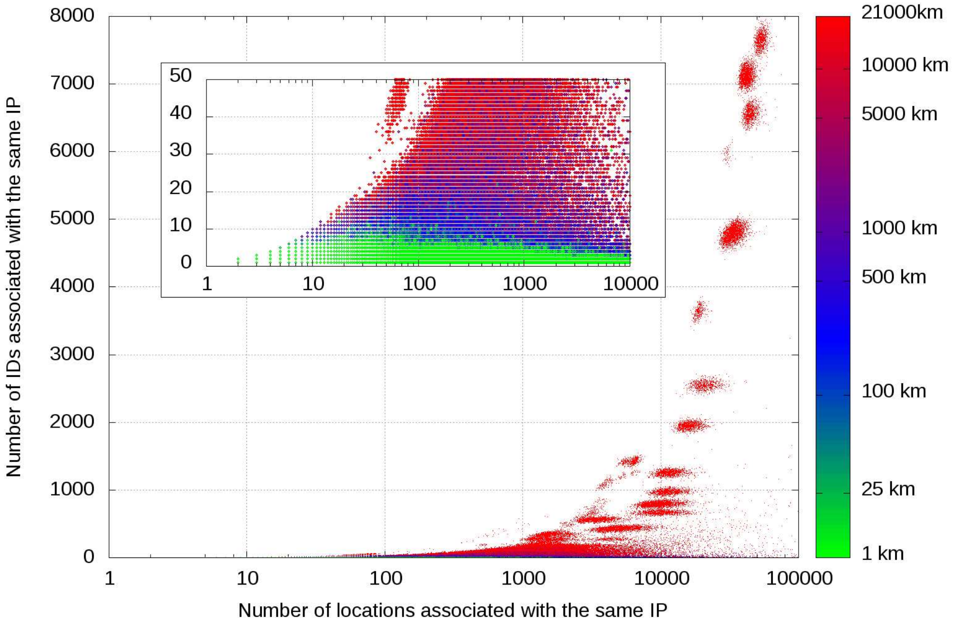


Figure 4.6: **Scatter plot of Locations and IDs with color coded maximum distance between any two locations for the same IP**

Figure 4.6 correlates the number of locations reported for an IP address (x-axis, log scale) with the number of user ID (y-axis) that were seen using the IP. Furthermore, each dot is colored with the maximum pairwise distance found in the distance set of the IP. If multiple IPs report the same location/ID count, the minimum of the distances is chosen. The small, embedded figure on the top-left show the same dataset, but with a zoom to the low-ID region. The x-axis of the plot has been cut off at 100,000 locations, as only a few IP addresses surpass that location count. The maximum locations associated with a single IP address are 3,701,423. Finally, all IPs with a location count higher than 100,000 have a max distance of more than $15,000km$.

The results shown in Figure 4.6 confirms that, as more devices share the same IP address, the maximum location spread drastically rises. However, the plot prefers short distances, hiding multiple samples for the same location/ID count with higher distances behind the smaller ones. This introduces a bias towards small distances. Thus, it can be concluded that IPs with a high ID count also have a high pairwise distance, while IPs with a low ID count might have a small distance.

An interesting observation about Figure 4.6 is the formation of clusters at the high location/ID region. Further investigation reveals that these all belong to ISPs offering mobile services. This observation raises the question what effect the ISP infrastructure has on the reliability of the end-user network position.

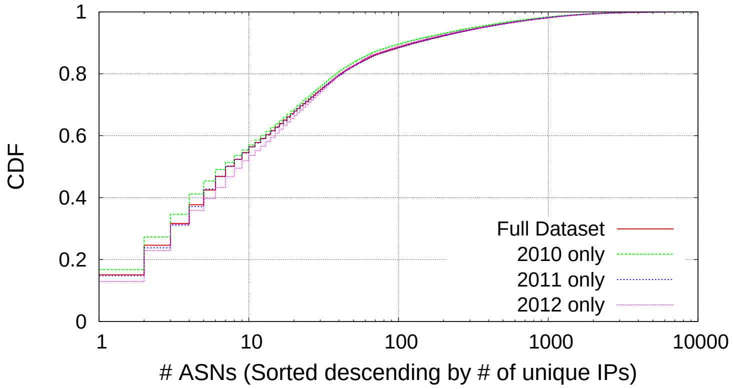


Figure 4.7: **CDF of the fraction of unique IPs per ASN (total and for individual years)**

4.2.3 Effect of AS Infrastructures

In order to quantify the effect of network infrastructures on the reliability of user-submitted GPS positions, we first turn to finding the bias towards specific ISPs. Remember that for the purpose of this evaluation we assume that an ISP runs exactly one AS. Figure 4.7 shows the CDF over all ASes ordered by the number of unique IPs they contribute. The largest network holds more than 15% of all entries, while the Top 10 ASes contribute more than 50% of all IPs. In total, there are 12,712 ASes in the dataset.

Turning to the evolution over time, we observe that the large ASes reduce their percentage over the years. While in 2010 the largest AS was responsible for 16.7% of all contributed unique IP addresses, this dropped to 14.8% in 2011 and 12.9% in 2012. Table 4.4 gives a more detailed view of the 10 largest contributors in terms of reported IPs. To no surprise, all major US-based ISPs, fixed and mobile, are present, including the large ones. Surprisingly, the most used IP address space belongs to the fixed lines of AT&T, while it is mainly mobile devices that are equipped with GPS. This means that end-users switch from the mobile network to wireless once this is available.

What cannot be seen is if the ordering in the ASNs changes and whether the absolute numbers are still growing. When splitting the dataset into years, we find that the top player, i.e. AT&T, increases the number of IPs by a factor of 1.5 from 2010 to 2011, while its overall percentage decreases. Thus, the MBS increases its total number of end-user subscriptions, while the percentage for the large ASes wanes.

Next, we analyze the impact of AS infrastructures on the reliability of the end-user submitted locations. Figure 4.8 again relies on the distance sets shown as CDFs of the maximum distance for each AS individually. We find two extremes in terms maximum distance distributions. On the one hand, AS 22394 has an almost perfect

Name	ASN	IPs	%
AT&T	7132	2,416,217	15.09
Verizon	19262	1,528,229	9.54
AT&T Mobile	20057	1,127,141	7.04
Bell South	6389	968,858	6.05
Cellco	22394	758,079	4.73
QWEST	209	698,667	4.36
COX	22773	530,113	3.31
Cable Vision	6128	359,225	2.24
Charter	20115	349,859	2.28
Windstream	7029	298,273	1.89

Table 4.4: **Top 10 ASNs in the dataset**

AS	7132	19262	20057	6389	22394	209	22773	6128	20115
single	17.4	16.6	31.4	20.2	21.5	21.7	14.2	26.9	28.1
0-dist	31.7	31.0	39.4	36.5	39.9	37.1	25.8	14.6	13.6

Table 4.5: **Top 10 ASNs percentages in single location IPs as well as identical positions**

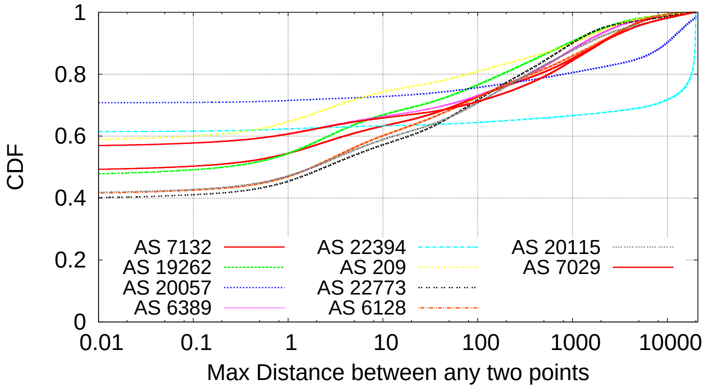


Figure 4.8: **Maximum location spread for IPs in the same AS (top 10)**

bi-modal distribution of locations, i.e. the distances are either very small or very large, with very few IPs between the two extremes. On the other hand, AS 22773 has a very promising distribution of maximum distances. Here, a third of its address space is has a distance larger than 0 but smaller than 25km, allowing for promising IP-Geolocation results.

Another observation is that all ASes show a large fraction of distances that are smaller than 10 meters. To check the cause for this, Table 4.5 shows the percentages of single location IPs as well as a 0-distance set. 0-dist is the percentage of IP addresses that report the same location multiple times, while single is the percentage of IPs that only appear once. Comparing the distribution of AS 22394 with Table 4.5, we conclude that almost all IP addresses for that AS either have perfect 0 distance (61.4%) or a distance greater than 1000km (33.4%), leaving only 5.2% of all IP addresses in the range between perfect 0 and 1000km. The same behavior, if not as drastic, is observed with the other large mobile AS in the dataset.

The second interesting feature about Figure 4.8 is that there are some ASNs that show a steady growth in distance. The most prominent in this case is AS 22773, where 40% if the entries report a single or 0-distance, while only 10% of the IP addresses report a maximum distance of greater than 1000km, leaving almost 50% of the samples in a range geolocation can potentially be trusted to confine the area in which the IP address is located.

When relying on end-user submitted locations for IP-Geolocation purposes, it is crucial to take the AS infrastructure into account. Especially for Mobile Providers, which tend to cluster their users behind proxies, the location of the aggregated end-users significantly differs from the network location in terms of geographical distance. Nonetheless, they use the same IP address. In contrast, some ASes, especially providers of home subscriptions, show promising results for reliable end-user locations.

4.2.4 Location Verification without Ground Truth

Not all IP addresses in the dataset are reliable in terms of geolocation. For example, when mobile users are aggregated behind proxies and gateways, their associated location become useless due to the device and the network location being far apart. These IP addresses need to be filtered from the dataset to increase the confidence in reported location being close to the geographical assignment of the IP address. To this end, we introduce the definition of a stable IP suitable for IP-Geolocation based on our experience with the dataset. Then we turn to the evaluation of our definition by comparing our stable IPs against Edgescape [12], a commercial IP-Geolocation database offered by Akamai. Finally, we use active measurements to further increase confidence.

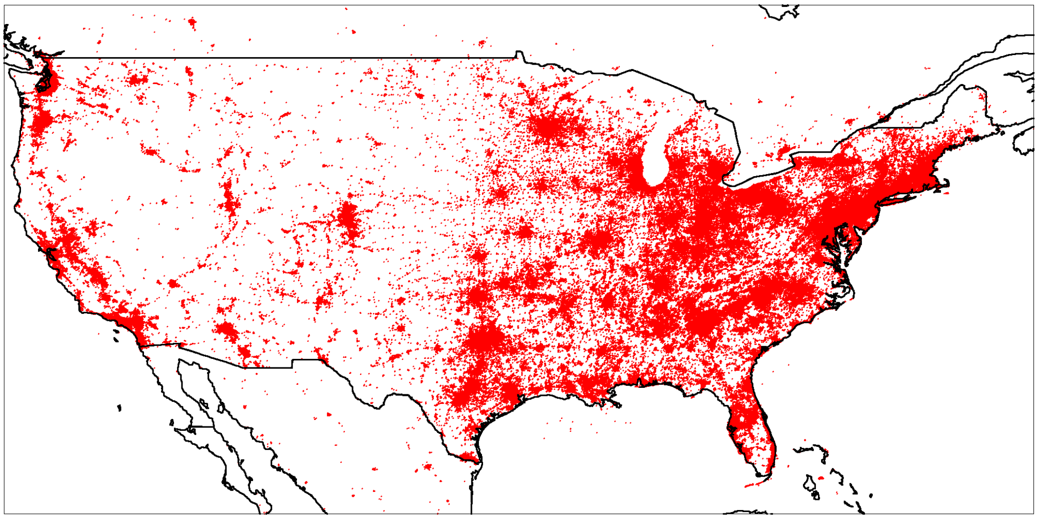


Figure 4.9: **Positions classified as Stable in the US**

4.2.4.1 Definition of a Stable IP Address

Our definition of a stable IP address is three-fold. The first part of the definition regards the number of samples. In fact, an IP address that only reports one location has no doubt about its location and no contradicting data to dispute it. But only one location is not reliable in terms of samples. Thus, we define that an IP address needs a minimum number of locations reported to be classified stable.

The second constraint is motivated by the temporal distribution of the samples. Specifically, it concerns the filtering of rapid connection attempts. When end-users connect to a few media streams in rapid succession and then never return, all locations are naturally the same, as the device did not manage to update. From this, we derive the requirement that an IP address needs to be sampled on different days. These days can be consecutive or with long periods of idle time in between.

The third requirement that an IP address has to meet regards the location confinement. This directly relates to the maximum distance between any two locations associated with the same IP. The smaller the maximum distance is, the better an IP address can be located. Thus, we require that the maximum distance cannot be higher than a certain threshold.

Finally, we need to take the duration of the dataset into account. Specifically, we need to consider the possibility of IP addresses being reassigned by the AS. Thus, for all IPs that meet the sample and temporal constraints, but fail at the distance, a simple clustering of IP addresses is performed to find location hot-spots. If clusters exist, and they are clearly separated in time, i. e., first one cluster is recorded, then another, each cluster is individually treated with regards to the other constraints. This leads to the fact that potentially two or more stable locations are made available per IP, each with a given duration of validity.

In order to apply the definition of a stable IP address, each of the three constraints needs to be set with values. To find these values, we rely on the observation regarding the mobile providers that aggregate users. In detail, we focus on the largest mobile provider in the dataset, AS 20057, and analyze its location counts, location spread and temporal distribution. Here, we find that effective settings to filter this AS are set at a sample count of 10 and the days at 5. We repeat the analysis with the second largest mobile provider, AS 22394, and find the same parameters fit this AS. When turning to the confinement constraint, we define a maximum distance of 25km, which is the size of the 10^{th} largest city, Nashville (Tennessee), in terms of area in the US. Thus, we choose these parameters to filter the entire dataset.

4.2.4.2 Applying the Stable Definition

Out of the 16 million (16,012,701) unique IP addresses found in the dataset, 1.8 Million (1,829,698) IP addresses matched our stable definition. Figure 4.9 illustrates the stable positions in the US. When comparing to the full dataset (i. e., Figure 4.2), it is clear that the stable definition removes almost all positions from obviously mobile positions. The most prominent feature is the absence of the two major highways in Alaska. The same effect can be seen throughout the rest if the country. Major roads are reduced to the towns alongside them.

The sampling of stable positions keeps the focus on the US. However, stable positions are also found in other parts of the world. These focus mainly on Europe, Japan and South Korea. This matches our earlier results, since these areas were the most represented countries after the US in the dataset. With respect to IP addresses being re-assigned to a different location, we find only a few examples where this can be observed. Thus, we conclude here that IPs we define as stable maintain their location for the entire time they appear in the dataset.

The next question that arises regards how well the stable definition manages to handle the different AS infrastructures. To this end, Table 4.6 summarizes the top 10 ASNs after the stability criteria has been applied and compares it to the ranking of the ASes from before the stability filtering. The first column gives the rank in the current ranking (total count of stable IP addresses) while the second and third column specify the AS name and the AS number. Next follows the total number of IPs that matched the stability criteria (column IPs) and the percentage of how many overall stable IPs (% total) this represents. Finally, the tables shows the percentage of stable IPs for the specific AS and the rank the AS had before the stability was applied.

	Name	ASN	IPs	% total	% stable	Old-Rank
1	AT&T	7132	237,905	13.0	9.8	1
2	Verizon	19262	224,482	12.3	14.7	2
3	Cox	22773	107,512	5.9	20.3	7
4	Charter	20115	65,838	3.6	18.8	9
5	Cable	6218	63,702	3.5	17.7	8
6	Bell South	6389	59,823	3.3	6.2	4
7	Qwest	209	51,385	2.8	7.4	6
8	RoadRunner	11427	45,945	2.5	21.6	13
9	RoadRunner	20001	45,005	2.5	21.1	12
10	Brighthouse	33363	42,532	2.3	19.6	11
26	Windstream	7029	16,990	0.9	5.7	10
109	AT&T Mobile	20057	797	0.044	0.07	3
394	Cellco	22394	88	0.0005	0.01	5

Table 4.6: **Top-10 ASes (in terms of IPs) before and after the stability criteria was applied**

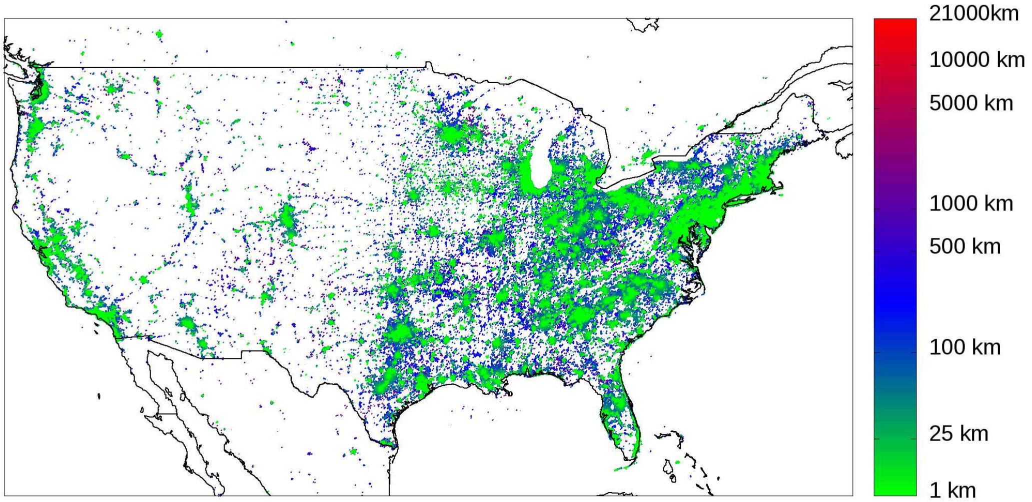


Figure 4.10: **Map of the distances between the reported user location and the geolocation database Edgescape**

In general, the top 10 do not change much. In fact, place 1 and 2 go to the same AS, which together now hold 25.3% (before 24.5%) of all IPs in the set. The ordering in the lower ranks mixes, but the general players are the same with the exception of the mobile networks. To no surprise, mobile networks dropped from the top 10 list. When looking at the top 100, we find that the only mobile provider registering more than 100 stable IPs is AT&T mobile, while all others drop to less than 100, and in some cases to 0 stable IPs.

Another interesting observation is the percentage of stable IPs. While some networks, e.g. Cox and Roadrunner, register more than a fifth of their IPs as being stable, others in the top 10 barely pass the 5% mark. For these networks, we find that the IPs are dropped from the stable set because they neither meet the temporal, sample or neither constraints. Only in very few cases is an IP address dropped due to distance constraint violation.

4.2.4.3 Comparing to an IP-Geolocation Database (Edgescape)

Next we turn to the geolocation database Edgescape and compare its information to user location. Edgescape is an IP-Geolocation database developed and maintained by Akamai. It uses several data sources, such as WhoIS, reverse DNS lookup, traceroutes and active measurements to infer the position of an IP address. Akamai's global server infrastructure plays an important role in this setup, as it allows for a diverse set of network measurement positions.

Unfortunately, Geolocation databases do not supply ground truth information. Thus, any comparison leaves the question of which location is the correct one, and, if they agree, whether the location is correct at all. However, if multiple databases agree on the same location, and this corresponds to the location that the user's device also supplies, there is a good chance that the location is correct.

We focus the comparison between Edgescape and the MBS on the location agreement for an IP. As an overview, Figure 4.10 plots the distance difference in the US. The position of each dot is at the end-user reported location, while the color indicates the distance to the location supplied by Edgescape. We focus the plot on the US due to most of the locations being reported from there. Again, we point out that the plot favors small distances, potentially hiding larger errors.

Most of the urban areas are marked with very low distances, indicating a close match between the coordinates from Edgescape and the MBS. Outside the dense urban areas, the distances increase moderately. However, in most cases the coordinates are less than 100km apart. While a 100km does not allow for a city match, it is close enough to agree on the state. Regarding rural areas of central USA, distances grow rapidly. At this point, Edgescape and the MBS disagree severely on the location of the users, sometimes with difference of over 5000km. In such cases, no information can be gleaned from the comparison other than the disagreement.

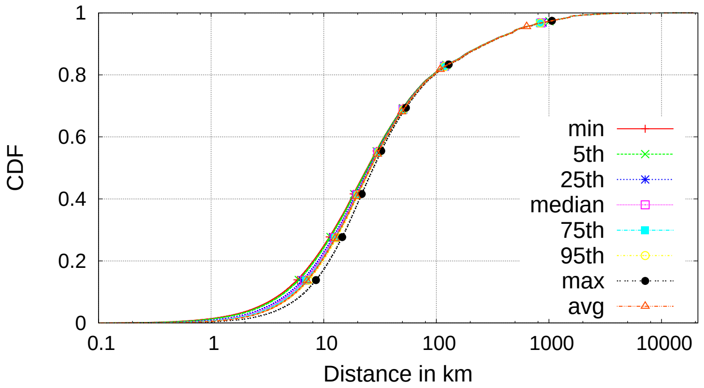


Figure 4.11: **CDF of quantiles comparing the distance between the end-user location and Edgescape**

Figure 4.12 shows the set of stable IPs on a global map, including zooms on Europe (bottom center) and Japan/South Korea (bottom left). Here, we observe the same trends as for the US. For Europe, almost all capitals show a very short distance. For Japan and South Korea the same trend can be observed, i.e. Tokyo and Seoul are clearly visible, with some additional locations around the major urban areas. Thus, we conclude that for urban areas with high speed, fixed Internet lines, the two datasets agree on location, while less populated regions tend to disagree.

As mentioned above, these plots favor close distances, which leads to the possibility of long mismatches in the urban areas not showing on the map. To confirm that long distances are not being systematically hidden by the shorter ones, the set of distances between the Edgescape and MBS are calculated. This is done on a per sample basis, meaning that for each time a location gets recorded in the dataset, this location is compared to the Edgescape position. From this, sets of distances are built, similar to the methodology introduced in Section 4.2.2.2. Since all IPs are classified as stable, they each have at least 10 samples and therefore at least 10 distances. From the distance sets for each IP, quantiles are taken and put into a CDF as shown in Figure 4.11.

All CDFs are tightly bundled, indicating that there are not many differences between the different quantiles. This indicates that the distance sets are quite narrow. With regard to Figure 4.11, 46.5 % of all stable IP addresses fall into this category. Also, 80.5% of all IPs have a distance of no more than 100km between the Edgescape and the dataset location. Finally, 17.8% of the IP addresses have a distance between 100km and 1000km, making an agreement on country level possible, while the rest (2.2%) has a distance of greater than 1000km, leaving only a continent or, in some cases, no clue at all where the IP address is located. With almost half of the IPs matching on a city level and more than 80% being close to it, we confirm that the two datasets agree in general and that large differences are not hidden.

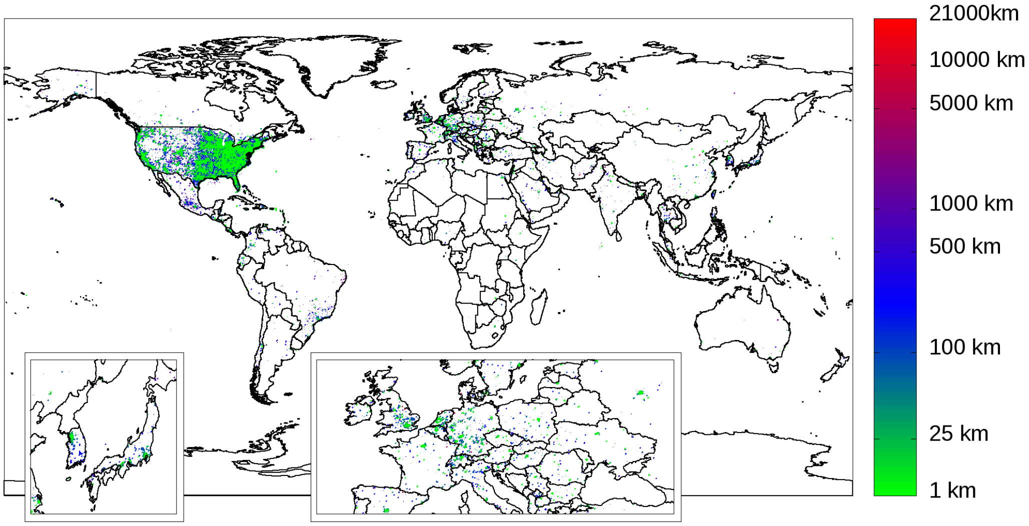
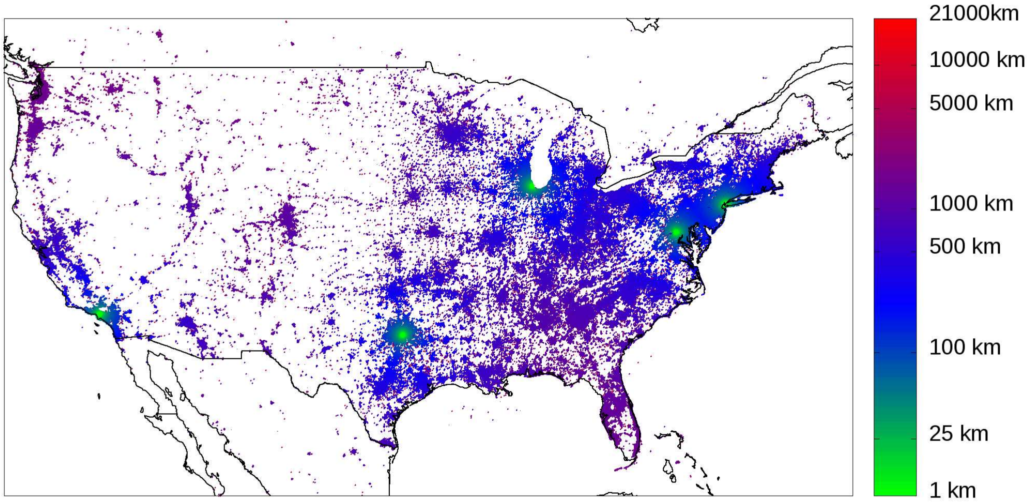


Figure 4.12: **World map with each stable position giving the distance to the positions reported by Edgescape**

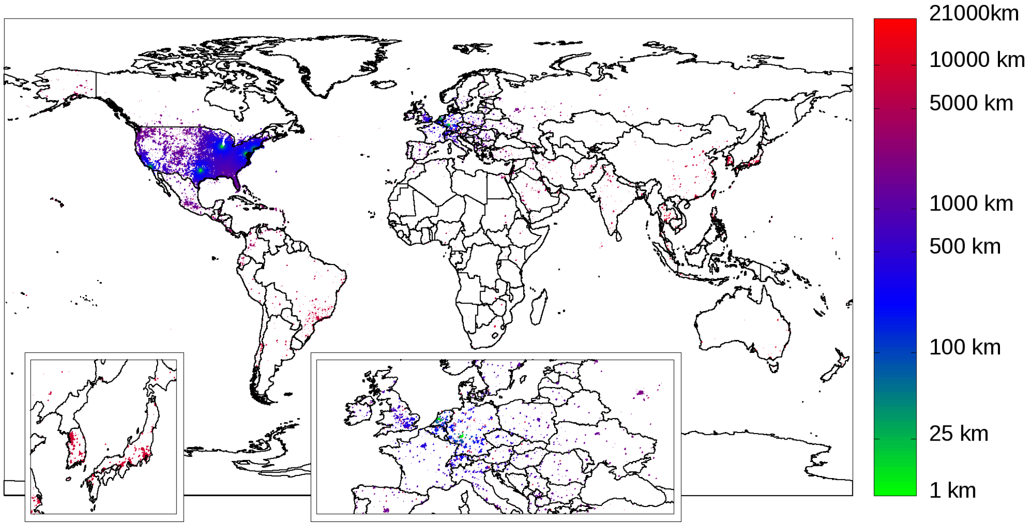
4.2.4.4 End-User to Server Assignment

Next, we turn our attention to the quality assignment of users to CDI servers, since close servers reduce delay, enable faster page-loading times and relieve the network of traffic. Under the assumption that the mapping of end-users to servers is perfect in terms of geographic distances, all clients should be connected to the server closest to them. Translated to geographical terms, this means that the servers are beacons, with distances growing longer as the user is farther away from the server. The locations of the servers are known to the MBS and geographical ground truth information is available. Figure 4.13a shows the distance between the user and the chosen server focused on the US. The server positions, i.e. New York, Boston, Chicago, Dallas and Los Angeles can be clearly seen on the map. Furthermore, the distances fan out in circles around these spots, indicating that, when a client is getting closer to a specific server, it is being mapped successfully to it. As with all plots before, this one also favors small distances. When looking at the samples more closely, we find that a significant portion of users are being directed to servers that are not the closest. In fact, it is not uncommon that end-users are assigned to servers several thousand kilometers away.

Turning to the global view of the MBS, we find two more locations from which users are being served: Frankfurt, Germany, and Amsterdam, Netherlands. Figure 4.13b shows the distance between the users and the servers around the world. The mapping in Europe is similar to the one in the US. With users close to the server, the distance between the two reported positions is also close. As the users are farther away from



(a) **US**



(b) **World**

Figure 4.13: **Distance between the end-user and the utilized server**

the server the distances also increase. But the amount of ”mismatched” servers in this case is significantly higher than before. In fact, for the non urban areas, there are substantial amounts of users mapped to servers on a different continent.

A special case is the far east region, i. e., Japan and South Korea. There are no servers in this part of the world. As expected, this leads to all locations reporting large distances to the server. This highlights the point that a deployment of the MBS in that part of the world would not only significantly reduce the distance to the servers, but would also take stress off of the network infrastructure.

A curious observation is that the Edgescape database and end-user locations in general agree on where the end-users are. Thus, the proximity of the end-user to the IP address is likely. In contrast, the end-user to server mapping does not show the same behavior, but even mismatches end-users where all three locations are close together. A possible reason for this can be that, while the locations are close, the network path between these locations is long.

4.2.4.5 Validation through Active Measurements

Our final analysis to verify the location of the stable IPs is by direct probing to find the correlation between geographical and network distance. In detail, we measure the time it takes for a probe to return to its origin. Today, this is usually done through sending pings and waiting for a reply [60, 73, 104, 137]. The assumption behind this practice is based on packets being bound by the speed of light. Thus, the maximum distance between the source and destination is bound.

Probe based measurements usually estimate the distance as being too large. This is due to middle hardware (switches, routers, queue buffers, processing delay, propulsion speed of the cables, transmission delay, etc.) that “slow” the packet down. Even in the case that all middle hardware does not induce any delay, any probe still travels along the cables and not the direct, shortest overland path. In the end, it is always the cable distance that is measured by network probes, not the geographical one. Nonetheless, correlating the geographical distance with the network distance allows an insight into the correlations between the two.

Given the extensive server deployment of the independent CDN that runs the MBS, it is a huge overhead to probe every stable IP address from every server. Therefore, the first question is about selecting the closest servers to be the source for the probe as only low round trip times bound the distance, while high values are useless. Thus, when choosing the probe source the supposed location is the one supplied by stable IP addresses. Another factor that needs to be considered is that pings are usually subject to high volatility due to the dynamics on the Internet. This leads to significant variations in the measurements due to cross traffic, temporary queuing on routers or, in the most extreme case, packet loss. Thus, multiple pings that are spaced over time need to be run.

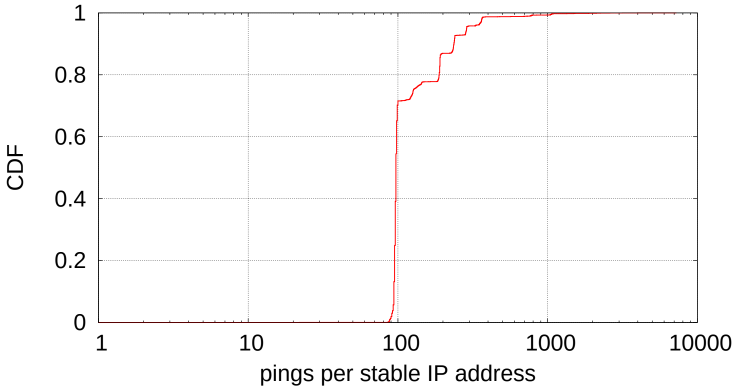


Figure 4.14: **CDF Pings per stable IP**

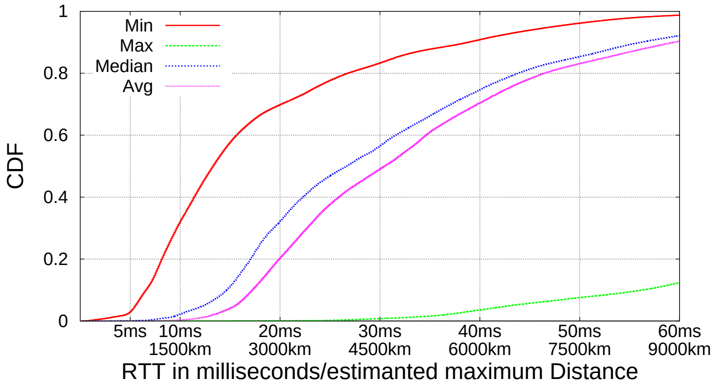


Figure 4.15: **CDFs of RTTs quantiles per stable IP**

We start our measurement by matching the extensive server infrastructure of the CDN that also supplied the dataset to the location of the stable IPs in order to find the 10 closest server sites. Each site selects one or more servers that is tasked to run 10 probes towards the IP address, while recording a timestamp as well as the round trip time. Unfortunately, it was not always possible to task exactly one server at a site, which sometimes leads to multiple servers running the probes concurrently.

Figure 4.14 shows the number of probes sent to each stable IP. We observe that 70% of the probes ran between 90 and 100 times. Unfortunately, due to maintenance and server load, some server sites were not able to perform the full cycle of 10 probes while others failed to select a single machine and ran the probes from multiple. Thus, some of the IP addresses were probed less than 100 times while others received significantly larger amounts. Nonetheless, more than 99.5% of the IPs were sent at least 80 probes. In total, 258 Million (258,575,859) probes were run over the course of two weeks. Out of the 1.8 million (1,826,805) IP addresses, 590 thousand (596,344) responded to the probes at least once.

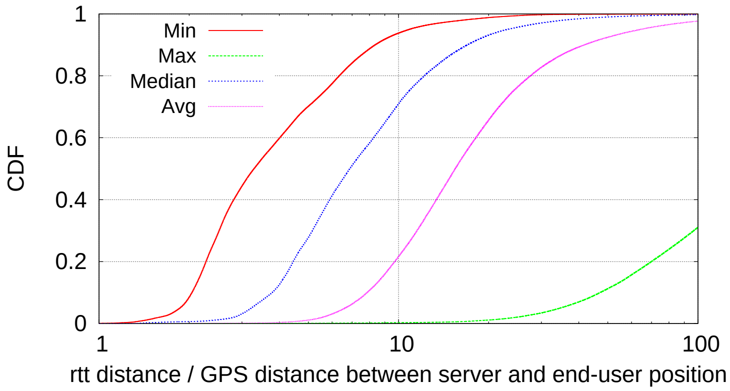


Figure 4.16: **Ratio between the Server/user Positions and the speed of light**

Next, we turn to the analysis of the probes. To this end, the minimum, median, average and maximum ping time was calculated for each responsive stable IP, see Figure 4.15. While the minimum ping time is reasonable, with 3% of the reachable stable IPs being below 5 ms (750km) and 32.3% (1500km) below 10 ms, there are also some samples which are not helpful. For example, any ping above 50 ms (7500km) makes it hard to even narrow down the continent the IP address is located at. If the minimum goes higher than this, no conclusion other than a slow network connection can drawn. When examining the median and average statistics, it can be observed that these show significantly higher round-trip times than the minimum. This is to be expected for the average, as it is dominated by large values. But the median is also significantly higher than the minimum and closer to the average. This illustrates that probes are highly volatile and advocates the need to perform extensive measurements before drawing any conclusion. The fact that the maximum ping time for almost every probed IP address is above 20ms (3000km) only adds to this observation.

Finally, we turn to the comparison between the geographical and network distance. Figure 4.16 shows the ratio between the two for the four minimum, median, average and maximum probe times. In an ideal world, where direct cables ran from each server to each stable IP address and the probes travel exactly at the speed of light, the ratio is exactly 1. At the same time, values smaller than 1 indicate that the supposed location is closer than a direct cable from the end-user to the server, while larger values indicate detours and middle hardware that ”slow” the probe down.

At this point, we only analyze the minimum round-trip time for each stable IP address further in order to estimate the distance. We observe that 0.008% (365) of the stable IPs are showing a value of less than one, indicating that the end-user is in fact closer to a server than anticipated. On the other hand, the rest of the IPs show a ratio greater than 1, with 8% being between 1 and 2, 36% between 2 and 3, 34%

between 3 and 5 and the remaining 22% being larger than 5. With the assumption that routers, queues and indirect cables should not add more than a factor of 3, it can be concluded that anything with a value between 1.5 and 3 (this is true for 42.8% of the IPs) is giving confidence that the location of the IP address has roughly the right distance to the server. Furthermore, we find that the correlation between geographical distance and network distance is distorted by the infrastructure, which increases the difficulty to map users to the right server.

4.2.5 End-User Submitted GPS Location Conclusion

Devices are increasingly equipped with the technology to infer their own GPS location. This is done through traditional GPS inference via satellites, or through assisted GPS (aGPS) which infers the location based on databases, nearby WiFi or other landmarks. This allows a device to specify where it is and potentially allows for improved CDI operation due to improved accuracy in determining the end-user's location. To this end, we analyze a trace from an independent CDN operating a Media Broadcast Service (MBS) spanning over two years with more than one billion connections supplying location information. The trace is biased towards the US mainland, but shows small usage patterns across the entire globe.

We find that the inference of network and geographical location can differ significantly. Specifically, we observe that, while land line subscriptions can in general be reliable when determining the location of a device, the mobile world paints a different picture. This is caused by the use of proxies and gateways which are common in mobile networks. Thus, our first finding is that user-based geolocation inference is only accurate in certain types of network infrastructures.

Driven by this observation, we introduce the notion of a stable IP, which is defined to be an IP address that meets three requirements: (a) it has a sufficient number of samples, (b) the samples are spaced over a longer period of time and (c) the area of the location is sufficiently confined. Applying this methodology to the dataset, we find that all mobile network operators are dropped, and only a small fraction (11.4%) of the IP addresses match the stable definition.

We then turn to the evaluation of the stable IPs and compare this to a geolocation database (Edgescape), the known server deployment of the MBS as well as active measurements targeted at it broadcast servers. We find that the stable IPs are in general matching all datasets, thus increasing the confidence that the these geolocations are reliable. However, when looking at the end-user to server assignment, we find a different picture. Here, we notice that a significant amount of the end-users are not being mapped to the closest server, in terms of geographical distance, but sometimes to servers several thousand kilometers away. In fact, our active measurements also show that the relation between geographical and network distance is a fragile one even for the IPs that are stable and operated on fixed infrastructures.

We argue that using IP-Geolocation based on end-user submitted locations is not helpful when mapping end-users to servers. This is especially true for mobile networks, where significant mismatches in the mapping occur, amplified by today's infrastructure which increasingly utilizes middle boxes.

4.3 Reliability of IP-Geolocation Databases

In order to widen the scope of IP-Geolocation, it is necessary to change the view away from the user submitted GPS locations and broaden the coverage of Internet address space. To this end, we turn our analysis towards IP-Geolocation databases. In detail, we are changing the approach from inferring locations through user submitted information and turn to evaluating database-driven geolocation in terms of reliability and accuracy. This enables an analysis of the entire Internet address space with the challenge that no alternative source for verifying locations is available.

Geolocation databases usually build upon a database-engine (e.g., SQL/MySQL) containing records for a range of IP addresses, which are called *blocks* or *prefixes*. Geolocation prefixes may span non-CIDR subsets of the address space, and may span only a couple of IP addresses. Examples of geolocation databases are *GeoURL* [53], the *Net World Map* project [98], and are provided as free [63, 68, 123] or commercial tools [11, 52, 62, 92, 108]. A known challenge of geolocation databases is that, besides the difficulty to manage and update them, their accuracy is more than questionable [59, 122], especially due to lack of information about the methodology used to build them. The crux of the problem is that prefixes within databases are not clearly related to IP prefixes as advertised in the routing system, nor to how those routing prefixes are used by their owners (e.g., ISPs, enterprises, etc). Indeed, even if many commercial geolocation databases claim to provide a sufficient geographic resolution, e.g., at the country-level, their bias towards specific countries make us doubt their ability to geolocate arbitrary end-hosts in the Internet. Few works focus on geolocation databases and their accuracy. Freedman et al. studied the geographic locality of IP prefixes based on active measurements [51]. Siwpersad et al. assessed the geographic resolution of geolocation databases [122]. Based on active measurements, the authors of [51, 122] showed the inaccuracies of geolocation databases by pinpointing the natural geographic span of IP addresses blocks.

4.3.1 IP-Geolocation Databases

In this analysis, we consider five IP geolocation databases. Two are commercial (*Maxmind* [92] and *IP2Location* [62]) and three are freely available (*InfoDB* [68], *HostIP* [63], and *Software77* [123]). Although these databases share some information about their construction processes, comments about how they are built are vague and technically evasive. As reported in [68], InfoDB is, for instance, built

Database	Blocks	(lat; long)	Countries	Cities
HostIP	8,892,291	33,680	238	23,700
IP2Location	6,709,973	17,183	240	13,690
InfoDB	3,539,029	169,209	237	98,143
Maxmind	3,562,204	203,255	244	175,035
Software77	99,134	227	225	0

Table 4.7: **General characteristics of the studied geolocation databases**

upon the free Maxmind database version, and augmented by the IANA (Internet Assigned Numbers Authority) locality information. The HostIP database is based on users' contributions. Finally, Software77 is managed by Webnet77, an enterprise offering Web hosting solutions.

Typically, a geolocation database entry is composed of a pair of values, corresponding to the integer representation of the minimum and maximum address of a block. Each block is then associated with information helpful for localization: country code, city, latitude and longitude, and Zip code. Table 4.7 shows the number of entries (i.e., the number of IP blocks) recorded in each database (column labeled "Blocks"). Most databases contain several millions of IP blocks. Only Software77 has much fewer entries: $99,134$. HostIP has the highest number of entries because it is composed exclusively of /24 prefixes. Compared to the more than $300,000$ prefixes advertised in BGP routing, one might be led to believe that the geographic resolution of the geolocation databases is much finer than the natural one from BGP routing [51].

To illustrate this point further, Table 4.7 provides the number of countries and cities retrieved from the database's locations. From the number of countries, we can infer that most of the world countries are covered. However, containing blocks for most countries does not imply that countries are properly sampled, neither from an address space perspective nor from that of geographic location. Figure 4.17 shows the cumulative fraction of blocks from the databases across countries. Note that countries on Figure 4.17 (horizontal axis) have been alphabetically ordered based on their ISO country codes.

Again, we stress that the number of countries represented in all databases gives the impression that they cover almost all countries in the world. This is misleading as more than 45% of the entries in these databases are concentrated in a single country: the United States (see Figure 4.17). The five databases display a similar shape of their cumulative number of blocks across countries. The big jump around country 230 corresponds to the over-representation of the U.S in terms of database blocks compared to other countries. Also, it is worth mentioning that the distribution of countries observed in WhoIS database (see Figure 4.17) exhibits the same behavior as geolocation databases. Returning to Table 4.7, we also notice the strong difference between the number of IP blocks and the number of unique (latitude, longitude)

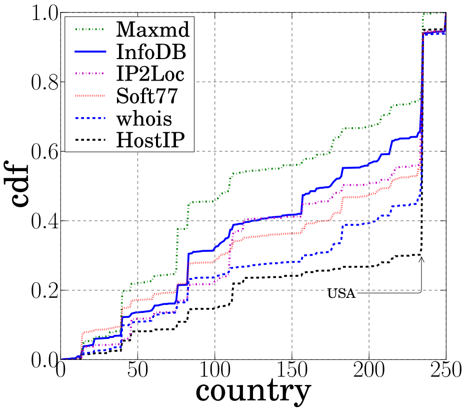


Figure 4.17: **Country distribution**

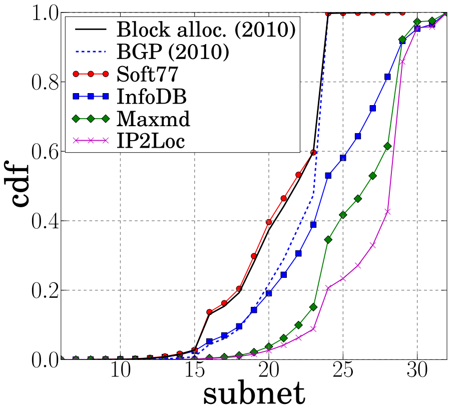


Figure 4.18: **Prefix distribution**

pairs. The perfect example of this is HostIP. While it contains roughly 8 million IP blocks, those blocks refer to only 33,000 (latitude, longitude) pairs. This observation casts some doubts upon the true geographic resolution of the databases.

4.3.2 Comparing Databases

Next, we investigate whether the construction of geolocation databases follows any global patterns. To this end, we focus on two aspects. First, we check how similar the blocks of the databases are to the address allocations and prefixes advertised in BGP routing. Second, we evaluate whether the construction of a database follows any demographic property, such as the amount of connected users there are in a given country.

Comparing the subnet size of database entries with those from the official allocations by the Internet routing registries and BGP routing tables is enlightening (see Figure 4.18). HostIP is not shown in the plot as it is exclusively made of /24 prefixes. We show results from February 2010, but it is worth noting that we observed similar results for other periods in 2009. Turning to the results, we notice that most allocated blocks and BGP prefixes are between /16 (65.535 IP addresses) and /24 (256 IP Addresses). Very few allocations and BGP prefixes are subnets smaller than a /24 subnet size. BGP prefixes are slightly more de-aggregated than the original allocations. Looking at the Software77 database, we notice that it is made of entries that have the same subnet size distribution as the original address space allocation. In fact, 95.97% of the entries in Software77 correspond to IP blocks as allocated in February 2010. As already expected when observing the block count, the other databases show a significant fraction of blocks smaller than /24 subnets. These databases split official address space allocations and BGP prefixes into finer blocks.

Prefixes advertised by BGP and allocated blocks could, however, constitute a first approximation to the databases' entries. Nevertheless, most of the IP blocks from Maxmind and IP2Location correspond to subnets smaller than /25 (128 IP addresses). In essence, Maxmind and IP2location entries substantially differ from BGP and official allocations by more than 50 % from a block size perspective. With such fine granular IP blocks, we expect a very high geographic accuracy. Again, because the way these databases are built is kept secret, we can only infer some of their characteristics. In particular, from these first observations, all the studied databases, except Software77, are clearly not related to official allocations and BGP routing tables. Even if the entries would closely match allocated or advertised prefixes, we would not expect that the locations attributed to them in the databases would be reliable. We believe this because the locations contained in the databases do not have to be related to how address space is actually allocated and used by its owners.

We base this argument on the fact that, while the Internet is a worldwide communication infrastructure, its deployment and usage differ significantly across different regions of the world. In the same way as address space allocation is biased towards certain regions of the world, geolocation databases should also reflect their usage. Thus, we now turn our attention to the countries that are present in all databases. A factor that is likely to explain the number of database blocks kept per country is the amount of *Internet users* per country, i.e., the number of people and infrastructure in a given country being connected to the Internet. The more popular the Internet in a given country, the more we expect to see entries in the databases for this country. The Internet users statistics we use are from 31st December 2009 [94].

We consider each country seen in the databases[1], and rank them according to the amount of people connected to the Internet (horizontal axis of Figure 4.19 in logarithmic scale). We then compute the fraction of blocks recorded in the different databases for each value of the number of Internet users and plot it in a cumulative way (vertical axis of Figure 4.19). We observe a strong relationship between the number of Internet users in a country and the importance of that country in the databases in terms of IP blocks. Countries with less than 1 million users are more or less non-existent.However, there is an exception to the general tendency drawn in Figure 4.19: a few countries with a large amount of population connected to the Internet are under-represented. The perfect example of this is China. While China has roughly 400 million Internet users, its database representation is a meager 1% to 5% depending on the database. Others examples are India, Japan, or Germany. The most represented country, U.S., is also one of the countries with the largest community of Internet users (roughly 230 million people). Finally, we cross-checked the amount of Internet users per country with the whole population of a country, leading so to an *Internet penetration rate.* In essence, more than 75% of the countries recorded in all the databases have a penetration rate higher than 0.6. The more

[1] Thus assuming that the country given in the database is correct for any given block.

popular the Internet among the population, the more frequent the country within the database entries. In summary, geolocation databases are therefore clearly biased towards regions with a high Internet usage. Again we note that HostIP is much more impacted than the other databases by the over-representation of the U.S. in its entries. This is expected since HostIP is based on users' contributions, which are most likely made by U.S. Internet users.

Given that geolocation databases are used for electronic commerce, we expect that they are engineered to target Internet users that are more likely to spend money on electronic transactions. Thus, the expectation is that the economic importance of a country is reflected in geolocation databases. We capture economic importance through the per capita *Gross Domestic Product* (GDP). We choose this measure because most economists use it when looking at per-capita welfare and comparing living conditions or use of resources across countries. Internet user statistics are again from 31$^{\text{st}}$ December 2009 [94].

Figure 4.20 is similar to Figure 4.19, but instead of Internet users, the horizontal axis shows the per capita GDP (in US dollars). In addition, we point out several countries (China, Italy, Germany, and United States) in the Figure. We observe a strong correlation between the number of prefixes in the databases and the per capita GDP. Indeed, countries with higher incomes have more opportunity to benefit from Internet services (Internet access, electronic commerce, online games, etc) than those with low incomes. As a consequence, it is not necessary for geolocation databases to keep many entries for countries having a low per capita GDP. With income, education and age being the principal factors determining the profile of Internet users, it is not surprising that countries with a low GDP are not well represented. Taking the example of China again, this observation is confirmed by the fact that most Chinese are living in rural areas, and thus are lacking computer skills or have "no need" of going online. Nevertheless, with the growth in the number of Internet users in China, one can expect a rise in the number of blocks related to China in new or revised geolocation databases.

Finally, we turn to the question whether the main contributing factor for the presence of prefix count in geolocation databases is related to the *Internet penetration* rate, i.e., the percentage of the population being connected to Internet. The more popular the Internet is in a given country, the more we expect to see entries in geolocation databases for this country. To this end, we compare the *Internet penetration* of a country to the percentage of blocks (frequency) this country holds in a geolocation database.

As before, the big jump observed in Figure 4.21 around 0.7 is due to the number of prefixes owned by the United States in all databases. Besides that, we observe a high correlation between the Internet penetration rate and the number of entries in the database. In essence, more than 75% of the countries recorded in all the databases have a penetration rate higher than 0.6. Put simply, the more popular Internet among the population, the more frequently the country is represented within the

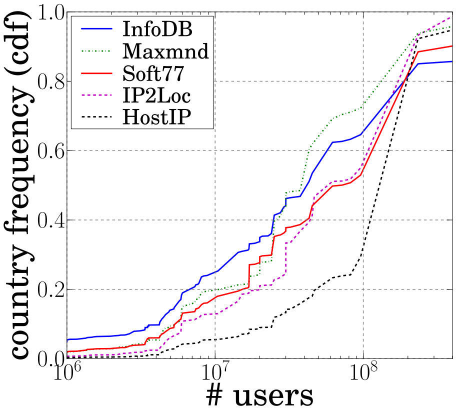


Figure 4.19: **Fraction of database prefixes as a function of Internet users**

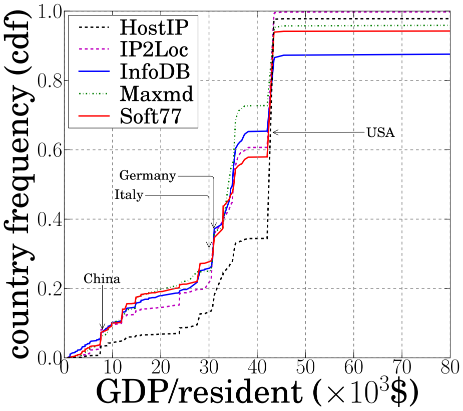


Figure 4.20: **Fraction of database prefixes as a function of per-capita GDP**

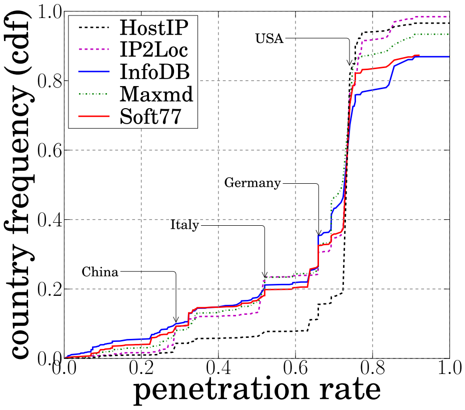


Figure 4.21: **Fraction of database prefixes as a function of Internet penetration rate**

DB1	DB2	Size
InfoDB	HostIP	19,481
	IP2Loc	5,213
	Maxmd	4,725
	Soft77	124
Maxmd	IP2Loc	2,701,034
	Soft77	84,469
Soft77	IP2Loc	85,577

Table 4.8: **Databases intersection size**

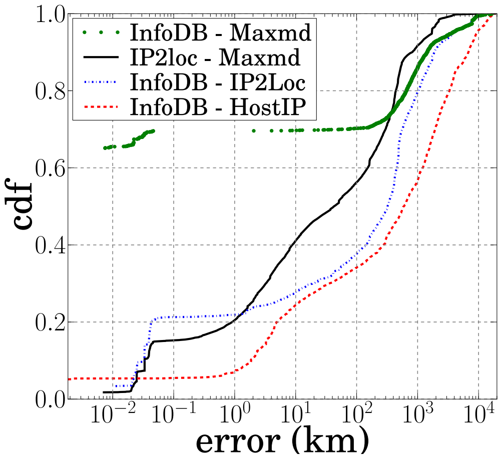


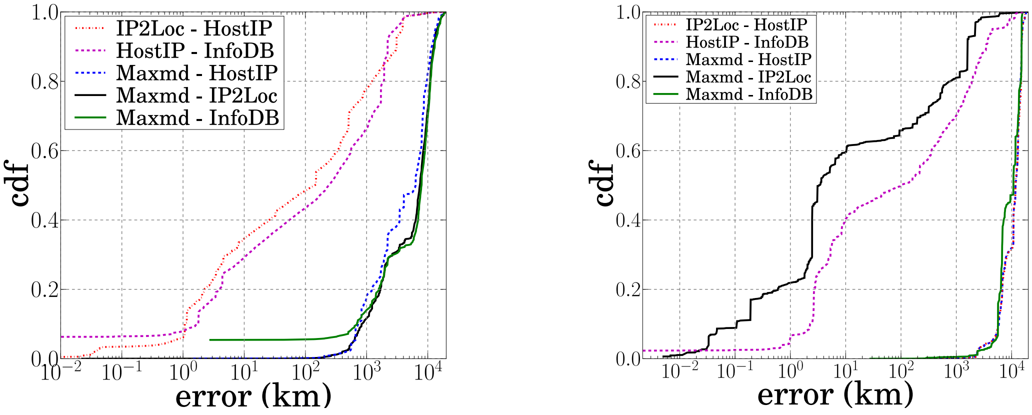
Figure 4.22: **Database differences for intersections**

databases entries. We observe that one country, the Falkland Islands, has an Internet penetration rate of 1, i.e. all users are connected to the Internet. Again we note that HostIP is much more impacted than other databases by the over representation of the US in its entries, which is driven by user submission from end-users.

4.3.3 Database Reliability

Next, we are interested to know whether geolocation databases are *reliable*. By reliable we mean that considering mutual comparison for a given IP address, the geolocation provided by the databases is the same (or very close). To this end, we perform an experiment based on a large set of randomly generated IP addresses. We evaluate to which extent the databases' answers would match when geolocating arbitrary IP addresses. We design and perform two kinds of experiments. First, we compute the intersection between each pair of databases, and verify whether the geolocation provided for the intersection is the same in the database pair. Second, based on a large set of randomly generated IP addresses, we evaluate to which extent the databases' answers would match when geolocating arbitrary IP addresses.

We begin this evaluation by examining the overlap that exist between the five studied databases. First, we observe the common entries that the databases share. The intersection has been computed by considering that two blocks match if they have the same starting IP address. As the distribution of block sizes strongly differ from one database to another (see Figure 4.18), requiring an exact match on the first and last IP address of a block would have led to a very small intersection size. Table 4.8 shows the size of the intersection between the databases. All intersections between the five considered databases that are not shown are empty. The largest intersection



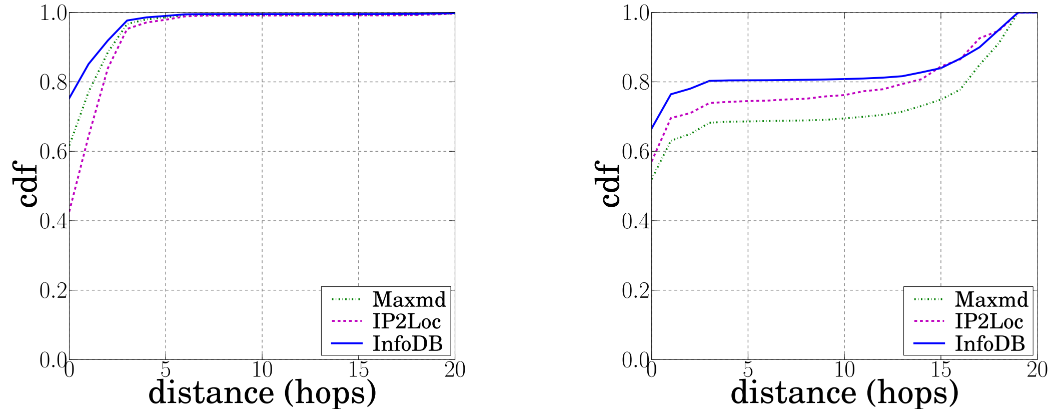
(a) **random IP** (b) **CDIs**

Figure 4.23: **Database discrepancies for randomly generated IP addresses and CDI infrastructures**

occurs between Maxmind and IP2location that share 2, 701, 034 IP blocks. This is more than 75% of Maxmind's number of blocks. The other pairs of databases share very few blocks. This observation hints at the possibility that IP2Location and Maxmind share similar methodologies to construct their databases' entries.

Next we turn our attention towards the evaluation about how these common blocks are localized by the databases, i. e., we want to understand whether common blocks also share common locations. Figure 4.22 depicts the CDF of the distance differences (x-axis, logarithmic scale) as returned by the pairs of studied databases for the common blocks. Note that Software77 has not been included in the plot as it only returns countries, but no GPS coordinate (latitude, longitude) pairs. The majority of blocks in common between Maxmind and InfoDB (65%) share the same localizations. This is expected since InfoDB is built originally from the free Maxmind database and augmented with other sources such as IANA assignments. However, the proportion of shared locations for other database pairs is very low. For instance, although they share a significant proportion of prefixes, IP2Location and Maxmind do localize only a tiny proportion of these common prefixes in the same locations (1.7%). Therefore, we conclude that even though their block selection methodology is quite similar, the process of assigning locations to the entries differs substantially. This suggests that the databases rely on very different location input and methodologies. In turn, widely differing methodologies with different results cast doubts on the ability of any database to accurately geolocate Internet hosts.

In order to better compare the geolocation databases for real world usage, we turn our attention to the second approach. Here, we consider the differences in geolocation databases when randomly sampling IP addresses across the available blocks. We randomly generate 10^6 IP addresses, each byte of an address being randomly selected between 0 and 255. We then geographically localize each of those



(a) **Orange** (b) **Level3**

Figure 4.24: **Distribution of Distances to the destination for DNS hints**

IP addresses using four databases: Maxmind, IP2Location, HostIP, and InfoDB. Then, we evaluate the difference between the locations returned by each database pair (in km), assuming that these locations are correct. Note that Software77 is not considered here as the number of recorded blocks is too small.

Figure 4.23a plots the cumulative distribution of distance difference (in km - x-axis in logarithmic scale) for the four considered databases. We notice first that a low proportion of IP addresses are identically geolocated by a given pair of databases. For instance, in only 5% of the cases, InfoDB and Maxmind provide the same answer. This is roughly the same proportion for HostIP and InfoDB. Figure 4.23a confirms that these databases disagree on a vast majority of IP address locations. In particular in all our comparisons more than 50% of the provided locations are at least 100km away from each other. Interestingly enough, locations returned by Maxmind exhibit the largest distance differences compared to other databases, with more than half of the sampled error distances larger than $7,000$km.

Finally, it is worth noting that we obtain very similar results to the random 10^6 IP addresses when using a set of $30,000$ IP addresses collected from various CDIs, as demonstrated by Figure 4.23b. This indicates that even a fixed infrastructure that is stable throughout a long period of time is geolocated at different positions by different databases.

4.3.4 DNS-based Assessment of Geolocation

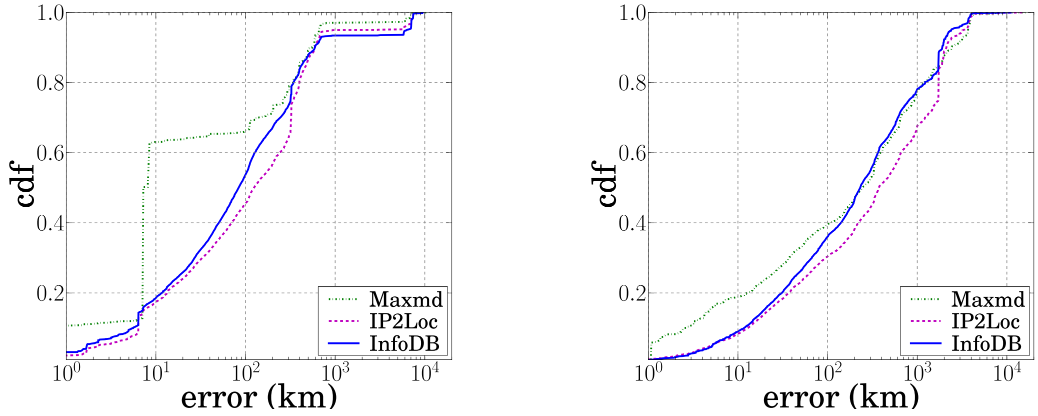
Obtaining ground truth information about how allocated address space is being used by ISPs is difficult, since it requires access to confidential information, e.g., IGP routing messages and router configuration. Without such an information, assessing the accuracy of geolocation database records can be done by carrying traceroutes towards a prefix and trying to locate the prefix using location hints from DNS names. Indeed, ISPs sometimes rely on naming conventions for their routers [126]. To this end, we select the address space advertised by two tier-1 ISPs, Level3 (AS 3356) and Orange (AS 3215), from a BGP dump from June, 30th 2010. We choose these ISPs because they carry hints about the location of their routers in their DNS names. All database records that belong to these BGP prefixes are searched in geolocation databases, leading to 347,736 IP blocks. For each IP block, we perform traceroutes towards the first IP address inside that block.

Next we run a DNS lookup for each IP address on the traceroute [2], starting at the closest to the traceroute destination and working backwards through the traceroute until a DNS query succeeds in resolving the IP address of the router on the path. As shown in Figure 4.24, in the vast majority of the cases, the hop with the DNS name we use to estimate the IP block's location is very close to the traceroute destination. In addition, in 66% of the cases, we succeed in resolving a DNS name. The DNS name returned is then searched for location hints. A location hint stands for a string that potentially indicates a city name. This is done by parsing the DNS name, looking for simple strings as done by the Undns tool [126], and then querying the Google maps service to find the likely location referred to by the hint. If Google maps returns coordinates and a name matching the hint for the location, we deem the IP block to be located close to these coordinates. If more than one suggestion is provided, or if no location hint is found in the DNS name, we simply discard the IP block. We have then been able to find 158 locations (double-checked manually), leading to a DNS-based estimation of the location for more than 165,000 IP blocks, i.e. 48% of the original blocks.

In summary, for each IP block, we selected the geographic coordinates of the router closest in hop count to an IP address from the IP block, as seen from the traceroutes and that returned a usable DNS name as the location. We stress that these considered locations are only estimations, and, in the event of them being correct, still add a likely uncertainty of tens of kilometers to the actual locations. However, these estimates can be good indicators of whether geolocation databases' returned locations are sufficiently close to an hypothetical ground truth location.

On Figure 4.25, we compare the distance inferred thanks to DNS hints, with the location provided by geolocation databases. Our results show that Maxmind performs well in the case of Orange (see Figure 4.25a), thanks to the high concentration of blocks on a few cities, e.g., Paris. Most of the blocks from Orange are located within 100km of the location inferred with the help of DNS. For Level3 (see Figure 4.25b), more than 60% of the IP blocks are mislocated by the databases by more than 100Km. Also interesting is that most of the Orange's blocks have location errors bound by the diameter of the country in which the ISP has most of its address space,

[2] A large fraction of ISPs do not provide any DNS names for their routers. The few who provide DNS names may not rely on an easily parsable location hint, e.g., the first letters of the city where the router is located.



(a) **Orange** (b) **Level3**

Figure 4.25: **Geolocation error of databases for Orange and Level3 based on DNS information**

which is less than $1,000$km. In the case of Level3, location errors are larger than $1,000$km for more than 20% of the studied blocks, which we attribute to the fact that Level3's main customer base is in the US, which has a much larger country diameter.

Based on location hints provided by DNS names, we measure the location mismatch, comparing our location with the one provided by the geolocation databases. By no means do our measurements allow us to make general claims about the accuracy of geolocation databases over the whole address space. More extensive measurements are necessary for this. However, given that the studied ISPs are mostly present in Europe and the United States, we cannot believe that the different ISPs we investigated are unfortunate cases where geolocation databases happen to provide poor accuracy at city-level and satisfactory accuracy merely at country-level.

4.3.5 Using ISP Ground Truth Data

Before we conclude our analysis of geolocation databases, we confront three of them with the network of a large European ISP for which we have ground truth about its allocated prefixes and their geographic locations. As compared to the earlier DNS analysis, this information is derived directly from the operation of the ISP network, removing the uncertainty in the position that was present when using DNS for the analysis. The IP's ground truth dataset has the same properties as the dataset used in Section 3.2 (Table 3.3). In fact both ISP information sets are taken from the same network, but at different times. We limit ourselves to IP2Location, Maxmind, and InfoDB because Software77 provides only a per-country localization and HostIP is limited to /24 blocks.

	Exact	Smaller	Larger	Partial
IP2Location	32,429	70,963	3,531	373
Maxmind	27,917	79,735	4,092	128
InfoDB	9,954	51,399	1,763	104

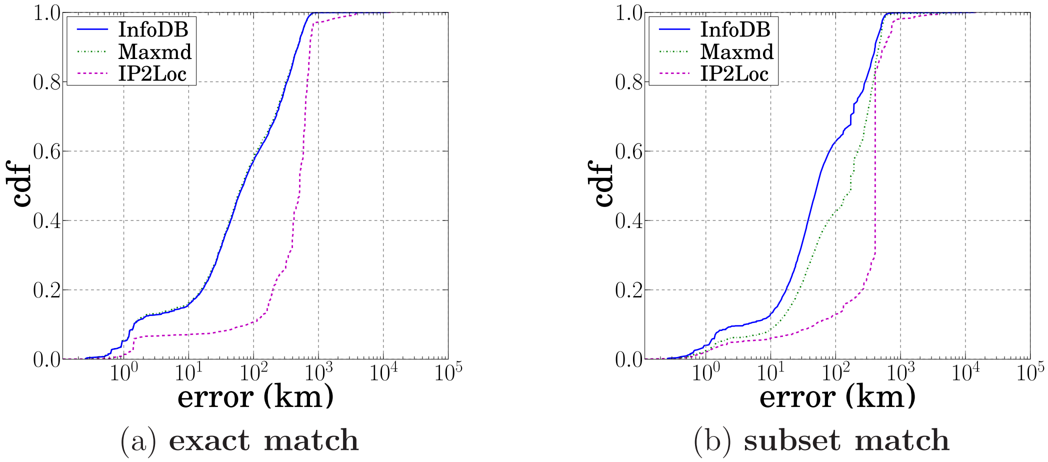
Table 4.9: **Matching prefixes from an European ISP against IP2Location, Maxmind and InfoDB**

We start the analysis of geolocation databases against ground truth information by extracting the internal routing table from a backbone router of a large European ISP. This dump contained about 5 million entries, describing more than $380,000$ prefixes (both internal and external). The reason for the large number of entries is the possibility that an ISP learns the same prefix from multiple, but different sources. From these prefixes, those originated by the ISP were extracted. This list was further trimmed down by dropping all entries not advertised by the ISP to external networks. This leaves us with 357 BGP prefixes advertised by the ISP which are reachable from the global Internet that can be matched against the databases. We call this set of internal prefixes reachable from the Internet the *ground_truth_set*, since we have POP-level locations for them.

Table 4.9 shows how the blocks of the three geolocation databases match the prefixes of the ISP (*ground_truth_set*). Four outcomes are possible for the match:

- *Exact* The prefix is present and has exactly the same size as the block
- *Smaller* The prefix is present but is smaller than in the database. This leads to aggregation of multiple different prefixes being aggregated into one block in the database.
- *Larger* The prefix is present but larger than in the database. This leads to over-splitting of the database, creating multiple entries where only one suffices.
- *Partial* The block from the database overlaps with two prefixes from the *ground_truth_set*. Neither the start nor the end of the block matches the beginning or the end of a prefix, and the block spans at least two prefixes.

The number of geolocation blocks that are smaller than prefixes from the ISP is almost as large as the full set of prefixes from *ground_truth_set*. Surprisingly, the databases also have prefixes that match exactly those from the *ground_truth_set* in about 40% (IP2Location), 34% (Maxmind), and 12% (InfoDB) of the cases. Databases therefore rely on the official allocations and advertisements from the ISP, but also try to split the blocks into more specific subsets for geolocation purposes, and do a reasonable job at it. Few blocks from the databases are bigger than those advertised by the ISP or partially match one from the ISP.



(a) **exact match** (b) **subset match**

Figure 4.26: **Geolocation error of databases for large ISP network with ground truth information**

The next step is to extract the city-level position of the routers advertising the subnets inside the ISP, giving us ground truth about the actual location where the prefix is being used by the ISP. To determine the exact location of the prefix, we rely on a passive trace of all IGP messages of one of the backbone routers of the ISP. Thanks to the internal naming scheme of the ISP, we obtain the GPS coordinates of the PoP in which each backbone router lies, and associate each prefix advertised on that router to the location of the router. These coordinates for each prefix are our ground truth used to assess the accuracy of the databases.

Figure 4.26 shows the distribution of the distances between the position reported by IGP and that reported by the databases, when looking at blocks of the databases that do exactly match (Figure 4.26a) or are smaller than prefixes advertised by the ISP (Figure 4.26b). The x-axis (in log-scale) gives a distance (in km) that we consider an error from the part of the databases, given the ground truth from the ISP. A value of 10 on the x-axis, for instance, shows the fraction of database prefixes that are less than 10km away from the ground truth.

From exact matches (Figure 4.26a), we observe that Maxmind and InfoDB have the same distance distribution to the ground truth (both curves overlap). This is due to the fact that InfoDB is based on the free version of the Maxmind database. Less than 20% of the exact matches for Maxmind and InfoDB are within a few tens of km from the ground truth. The rest of the blocks have errors distributed between 10km and 800km. Note that 800km is the maximal distance in the country of the considered ISP. IP2Location has much larger errors than Maxmind and InfoDB for the exactly matching blocks, with errors ranging between 200km and 800km.

For databases, blocks smaller than the ISP prefixes (Figure 4.26b), we observe two interesting behaviors. First, InfoDB and Maxmind have different error distributions, with Maxmind actually performing worse than InfoDB. This is unexpected given that InfoDB is based on the free version of Maxmind. The explanation has to do with the commercial version of the Maxmind database splitting prefixes from the ISP into very small blocks, many containing only eight IP addresses. Splitting is intended to improve the accuracy of the geolocation, but turns out to make geolocation worse, given that many small blocks have incorrect locations. The second observation we make from Figure 4.26b is the big jump for IP2Location around an error of 400km for about 50% of the blocks smaller than the ISP prefixes. By checking those blocks, we notice that these belong to a few prefixes from the ISP that are advertised but are partly unused. These large prefixes are currently advertised from a single location in the ISP network. A large number of database blocks consistently mislocate subsets of these prefixes.

Finally, we report a high success rate in providing the correct country of the considered IP blocks (between 96% and 98 % depending on the database). We conclude that some databases actually do a decent job geolocating some of the address space of the ISP. In most cases, however, the location given by the databases is off by several hundreds, even thousands of kilometers. Furthermore, by trying to split the address space into too small blocks, the databases make mistakes that are hard to detect unless one relies on ground truth information from the ISP that owns the address space. To conclude this section, we cannot trust the databases for the ISP at the granularity of cities, especially given the large relative errors they make compared to the span of the considered country (800km). Their country-level information however seems globally accurate. Our findings here confirm our DNS based conclusions from Section 4.3.4 regarding Orange and Level3.

4.3.6 Geolocation Database Conclusion

Our findings indicate that geolocation databases often successfully geolocate IP addresses at the country-level. However, their bias towards a few popular countries, those mostly those having a large number of Internet users, makes them unusable as general-purpose geolocation services. We observe significant differences among the locations they return for a given IP address, often in the order of hundreds of kilometers. Our results based on ground truth information from a large European ISP show that the databases perform poorly on the address space of the ISP. One of the reasons we identify for their poor geolocation abilities is the way databases try to split prefixes advertised by the studied ISPs into very small blocks. Instead of improving the geolocation accuracy, significant errors are introduced for a large number of blocks, especially at city-level.

4.4 Summary

IP geolocation has several applications today, including advertisement, localized services or optimizing CDI operations. In this study of geolocation, we focus on two aspects. First, we look at it from an end-user perspective to evaluate if geolocations from end-devices can be trusted to reliably identify a network position. We focus our effort on a server log from a large independent CDN spanning multiple years. Furthermore, we define the notion of a stable IP address in terms of area, duration and samples. Equipped with this definition we find that the ability to correlate an IP address with a geographic location heavily depends on the ISP itself. While ISPs that mainly run land line subscriptions show results that are not completely discouraging, mobile subscription ISPs are not showing any correlation between the reported user position and the used IP address. This is due to the fact that, in mobile networks, the CDI servers need to be close to the gateway or proxy instead of being close to the user.

Our second analysis compares multiple commercial geolocation databases with each other and evaluates their performance in an ISP we have ground truth for. The first key observation here is that geolocation databases are biased towards countries with high Internet penetration. The second observation is that, while the IP geolocation databases manage to determine the country an IP address is in most of the time, their city level resolution is questionable. In fact, when comparing the Geolocations to our ground truth dataset, we find that half the IP addresses are mislocated by at least half the country size.

In light of these results, we conclude that IP-Geolocation, either from end-users or from databases, is not fit to further improve the CP operations by supplying superior, close servers than network measurements do. This result motivates us to look for a different solution to enhance the CP operations while helping ISPs to better handle CDIs' volatile traffic.

5

Enabling Cooperative Content Delivery

Content Distribution Infrastructures (CDIs) today face several challenges (see Section 2.3), including end-user location (see Section 4.2.4.3) and performance optimization(see Section 3.2.2) as well as difficulties regarding costs of content-delivery and detection of network bottlenecks. This is complicated further by the vast infrastructure they operate and the ever increasing demand for content (see Section 3.1). Given that a substantial fraction of the overall traffic is available at multiple locations, sometimes even inside the ISP requesting the content, and that there is significant path diversity to the servers hosting the content, we propose a system, named **Provider-aided Distance Information System (PaDIS)**, that allows for CDIs to use the ISP's intimate knowledge of the network for server selection optimization. PaDIS uses the insights from Aggarwal et al. [9, 10] on collaboration between P2P networks and ISPs and evolves them to be applicable to CDI traffic.

At its core, PaDIS' task is to act as an interface for the CDI and is run by an ISP. There, PaDIS collects and processes information about the network in real time, while making this information available to the CDI through a recommendation service. More specifically, a CDI queries PaDIS by submitting a list of possible IP addresses and a source. In the case of a CDI, the list of IP addresses are the servers that are eligible to deliver the content, while the source is the end-user requesting content from the CDI. Upon receiving such a list, PaDIS will rank the submitted IP addresses according to its metrics, e. g., distance within the Internet topology, path capacity, path congestion, path delay, etc. Once the list has been reordered, it is sent back to its origin. Through this mechanism, it becomes possible for CDIs to take the state of the network into account when selecting servers for end-users.

In this section, we first outline the motivation for each involved party for deploying PaDIS, then turn to its design as well as how and where it fits into today's Internet architecture. Then we turn our attention to the question of how PaDIS can be used to take advantage of server diversity for content delivery, before discussing the scalability and responsiveness properties of our prototype implementation.

5.1 Incentives for using PaDIS

The introduction of a new service raises the question of what individual parties gain from deploying it. As PaDIS is no different in this regard, it needs to motivate both CDIs and ISPs to implement and use it. In order to do so, PaDIS' design has its focus on offering an operational advantage from the first collaboration onwards while keeping as much information as possible confidential.

5.1.1 CDIs

The CDIs' market requires them to enable new applications while reducing their operational costs and improve end-user experience [102]. By cooperating with an ISP, a CDI improves the mapping of end-users to servers, improves in the end-user experience, gains accurate and up-to-date knowledge of the networks and thus increases its competitive advantage. This is particularly important for CDIs in light of the commoditization of the content delivery market and the selection offered to end-users, for example through meta-CDNs [38]. The improved mapping also yields better infrastructure amortization while the need to perform and analyze voluminous measurements for network conditions and end-user locations is removed.

5.1.2 ISPs

ISPs are interested in reducing their operational and infrastructure upgrade costs, offering broadband services at competitive prices, and delivering the best end-user experience possible. A better management of traffic by collaboration with CDIs through PaDIS gives ISPs a competitive advantage in terms of customer satisfaction. Furthermore, cooperation with a CDI offers the possibility to do global traffic and peering management through an improved awareness of traffic across the whole network. For example, peering agreements with CDIs can offer cooperation in exchange for reduced costs to CDIs. This can be an incentive for CDIs to peer with ISPs, and an additional revenue for an ISP, as such reduced prices can attract additional peering customers. Furthermore, collaboration with CDIs has the potential to reduce the significant overhead due to the handling of customer complaints that often do not stem from the operation of the ISP but the operation of CDIs [24]. Through this, ISPs can identify and mitigate congestion in content delivery, and react to short disturbances caused by an increased demand of content from CDIs by communicating these incidents back directly to the source.

5.1.3 Effect on End-Users

Collaboration between ISPs and CDIs in content delivery empowers end-users to obtain the best possible quality of experience. As such, this creates an incentive for end-users to support the adoption of collaboration by both ISPs and CDIs. For example, an ISP can offer more attractive products, i. e., higher bandwidth or lower prices, since it is able to better manage the traffic inside its network. Also, thanks to CDI being aware of network conditions, ISPs become more attractive to end-users due to increased performance. This can even be done through premium services offered by the CDI to its customers. For example, CDIs delivering streaming services can offer higher quality videos to end-users thanks to better server assignment and network engineering.

5.2 PaDIS Architecture

Figure 5.1 shows an overview of PaDIS architecture. It is split in two general parts: answering queries from clients who want network-based ranking decisions (*Query Processing*), as well as to build and maintain the Network view *Data Management*. The former part consists of the `Query Processor` and the `Location Ranker`. Its main design goal is to efficiently answer queries from clients, in order to maximize the throughput and minimize the delay of the system. The latter part includes collection and processing of network information, such as topology, routing, IP address assignment, link capacity, link congestion, and link delay. Furthermore, custom attributes can be supplied by the network administration to augment the network map. To be real-time capable, it needs to keep in sync with the network at all times, e. g., the network status is continuously fed into the system. Finally, all ranking decisions are are used as an input to the data management to enhance the Network Map even further. Internally, PaDIS represents the network as an annotated, abstract and network technology independent graph, which is stored in the Network Map Database. The Network Map Database is the core on which PaDIS bases all of its ranking decisions.

5.2.1 Data Management

The data management subsystem consists of components for information retrieval and monitoring, the network map database and the Frequent Hitter detection.

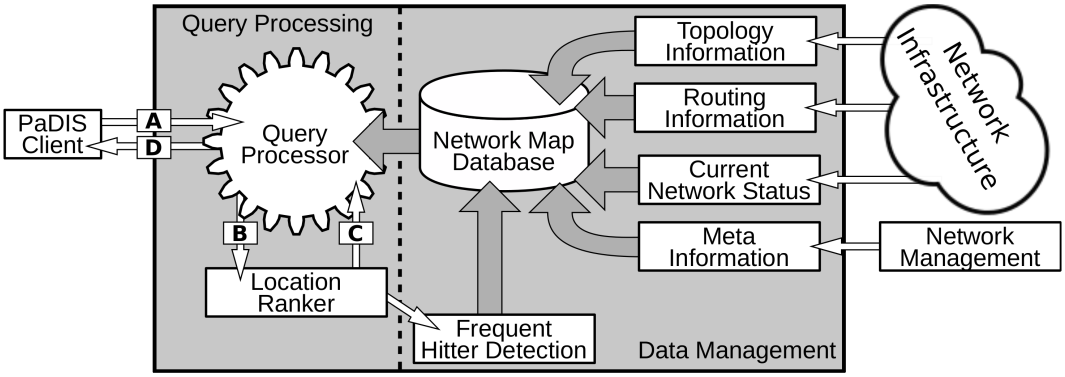


Figure 5.1: **PaDIS Architecture**

The information retrieval is split into multiple parts for high flexibility because different networks use different technologies in their network infrastructures. By splitting PaDIS into individual parts, each becomes easily replaceable, what in turn allows PaDIS to adapt to the different network environments effortlessly. In Figure 5.1, four modules of network monitoring are shown, i. e., Topology Information, Routing Information, Current Network Status and Meta Information that feed information into the Network Map Database directly and automatically from the Network. Furthermore, the Meta Information module is optional to add information not available from the network, while the Frequent Hitter Detection is a feedback mechanism to the database from the requests processed by Query Processing. The following list illustrates the modules' different tasks and how and why they gather information.

- **Topology Information**
  This part is responsible for gathering the basic network topology as seen by the technology of the Network Infrastructure. To this end, it uses a passive listener to gather the Interior Gateway Protocol(IGP) messages being flooded in the network. The most commonly used protocols are OSPF [31, 96] and IS-IS [103]. But to not limit PaDIS' scope, additional protocols (e. g., MPLS [111]) for network management have to be considered when discovering network topologies; adaptivity is of utmost importance. By encapsulating Topology Discovery into its own module, it is possible to create one module for each protocol (and their variants due to differences in vendor implementations) without having to change the architecture of the overall System.

  One limitation arising from this method of topology discovery is the availability of global network state. PaDIS is designed to know a network's topology, and by extension, it can only work with protocols that use a link-state approach. If global knowledge is not available in the network, as is the case in RIP, PaDIS cannot discover the topology through IGP.

Once the technology used by the Network infrastructure is known, the Topology Information System builds an adjacency to one of the routers in the network using the specific protocol and gathers all information available. Most commonly, the information about the network is in the form of protocol-specific Link-State-Advertisements (LSA). Once all LSAs have been gathered from the network, they are parsed, and the information is converted into general state information that fits the abstract network map, where it is then fed.

- **Routing Information**
  The Routing Information's task is to augment the topology with information about the assignment of IP addresses and subnets. Usually, the topological information learned from IGP does not encompass the specific assignment of IP addresses inside the network as supplement IP information can easily overload it. In order to learn this assignment, every router uses the Border Gateway Protocol (BGP [110]). Also contrary to IGP protocols, BGP is the de-facto Internet standard, meaning it is the only protocol used throughout the entire Internet for transporting IP assignment and routing information.

  When using BGP inside an ISP, it builds a full mesh of connections between all BGP enabled routers. However, with large networks of multiple thousands of routers, this induces a huge overhead in terms of connections. To this end, BGP implements a two level hierarchy of routers, divided into the edge and the core. Each BGP router on the edge has one connection to one or more core routers. More importantly, the edge routers lose full visibility of the global information, as they are only responsible for passing on learned information while retaining only information that they themselves need. On the other side, core routers called Route Reflectors, rebuild the full mesh with each other. Also, all information learned from any other BGP router, be it core or edge, is stored there. Thus, BGP Route Reflectors have the full visibility of all IP assignments within a network.

  Not all IP assignments happen inside an ISP. The Internet is a network of networks, illustrating the fact that other networks also assign IP addresses to their infrastructure. In order for those to be reachable, this information needs to be communicated to neighboring networks. Along the same lines as the internal prefix propagation, BGP is again used to carry this information across the borders between different networks. And as before, edge routers in an ISP pass all information along to the Route Reflectors, while keeping only the information necessary for themselves.

  In order for PaDIS to be able to gather a full view of the network and the IP assignment inside a network, it needs the full BGP information. Thus, by connecting PaDIS to one of the BGP Route Reflectors, it is able to learn all IP related information. It again parses and converts this information into the internal, generic format used by PaDIS and communicates the IP assignments to the Network Map Database.

- **Current Network Status**
  While the basic information about the path of the network is supplied in the two modules already discussed, the Current Network Status module is about the real-time capabilities of PaDIS. Its job is to supply near-realtime information about link utilization, latency, capacity changes, etc. to the Network Map Database. The more information can be gathered with this component, the richer the Network Map Database can be annotated, leading to more fine grained decisions in the Location Ranker.

  However, gathering the status of a network is a very challenging task, due to the fact that there is no standard technology for this. While most networks today already implement a technology to watch their network status on a real time scale, it is unclear how and what information is available. Possible ways collect network information are through router vendors directly if this is supported, through third party applications (Cariden MATE[28]) or through self build solutions that are unique to the network. Nonetheless, almost all network operators are capable of monitoring their networks today.

  Therefore, the Current Network Status module of PaDIS has to be the most adaptive part of the architecture. Since there are dozens, if not more, solutions out there to collect network information, it is necessary to abstract this variety away from the Network Map Database in order to maintain consistency. This, in the first step, means interfacing with a (possibly) unique system, and then recovering and converting the information needed for PaDIS to be able to use it.

- **Meta Information**
  As shown in Figure 5.1, the Meta Information in PaDIS is the only component not being fed directly by the Network. Instead, this module is about adding non-technical attributes to PaDIS. These can range from the monetary importance of a specific link over the general infrastructure cost on a per link basis to contractual constraints negotiated with users of PaDIS.

  Thus, this module is individually suited to each network PaDIS is being run in. Its task is to annotate the graph maintained by the Network Map Database with extra information that can be used by the Location Ranker. None of the information supplied by this module is needed for the basic operation of PaDIS, making this a purely optional component of PaDIS.

- **Frequent Hitter detection**
  The task of the Frequent Hitter detection is a feedback loop inside PaDIS that aggregates information on what the Query processing has been doing. It is based on probabilistic lossy counting [33] and keeps track of statistics, ranking and requests. For example, it keeps a list of the top-n IP addresses requesting rankings, the top-n IP addresses ranked top of the list and the top-m most used links when selection path. This information is regularly fed to the Network

Map Database in order to make it available to to the Location Ranker and, through this, these statistics can be used as input for a ranking function itself when processing new requests.

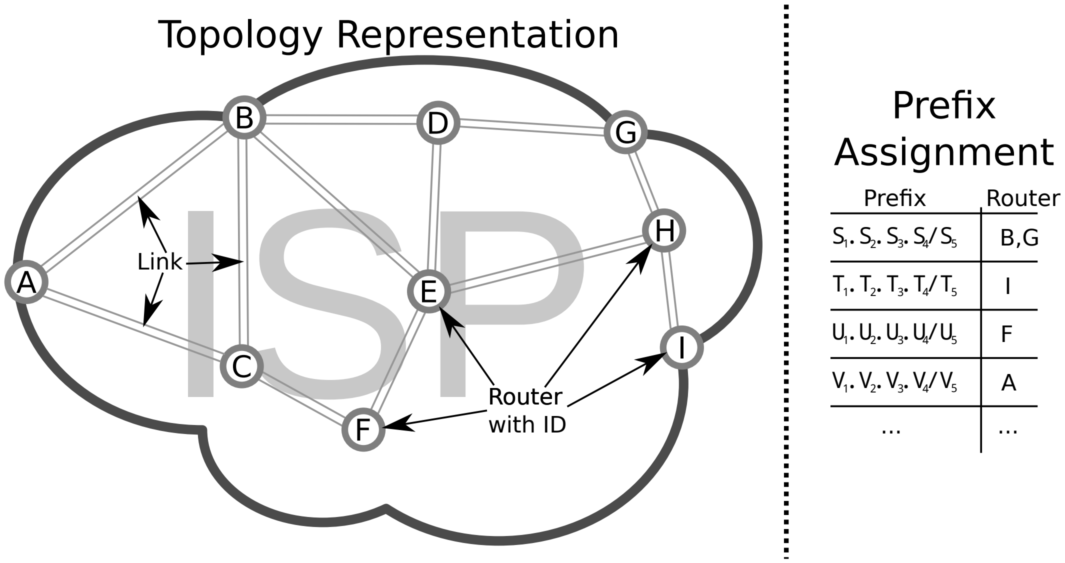
The **Network Map Generator** is responsible for maintaining an up-to-date view of the topology supplied by the modules feeding information. It has no active component in terms of querying for information. It uses specific data structures in order to store information efficiently and to maximize the lookup speed of path information.

When looking at information being given to the Network Map Generator by the Topology and the Routing Information, it becomes clear that they operate on very different timescales. More specifically, the Topology information is unlikely to change fast or drastically. Of course, there are changes to the configuration, and sometimes there are link and/or router failures. But, the topology information in itself is a very slow paced process that usually does not see more than a couple of changes a day, if any. On the the other hand, if something changes, major re-calculations have to be undertaken, as it is unclear how many paths have been affected by the change. In practice, a change in the topology is, at worst, a breadth first search from every node in the graph to every other node ($O(log(n)n^2)$ where n is the number of nodes). In contrast, the Routing Information is a medium-paced source of information, meaning that there can be several changes per minute. Furthermore, since most of the updates in the Routing Information are learned from other networks, there is no control of the network operators over these changes either. However, updating the routing is, if it is not coupled with the topology, a very fast operation ($O(1)$) since it only requires an update of the subnet to router mapping.

Exactly for this reason, we design the Network Map Generation as a split architecture between the topology and the IP assignment. The graph representation of the network (Figure 5.2 left) being derived from the topology information does not contain any IP address assignment. Each node (router) in the graph has an ID assigned to it, while the links are identified by the nodes they are attached to. Also, paths are calculated according to the weights configured IGP weights and are agnostic to any IP address or subnet assignment.

To map the IP addresses and subnets into the topology, a second data structure is used (Figure 5.2 right). This data structure is designed for fast lookups of IP addresses, implementing the longest prefix match in a patricia-trie [76]. Once the best matching prefix has been found, the data structure returns the node-ID this prefix can be found at. This allows for IP addresses and prefixes to be reassigned in the network, independent of the network topology.

Furthermore, splitting the two data structures lowers the memory consumption of PaDIS. In the real network, each router holds a full routing table to all prefixes that are known and advertised. This is potentially the address space of the entire Internet. Without the split between the topology and the subnet assignment, each



Prefix	Router
$S_1.S_2.S_3.S_4/S_5$	B,G
$T_1.T_2.T_3.T_4/T_5$	I
$U_1.U_2.U_3.U_4/U_5$	F
$V_1.V_2.V_3.V_4/V_5$	A
...	...

Figure 5.2: **Split between Topology and Routing information in the Network Database**

node in the graph needs to hold a full routing table to each prefix. In contrast, in the chosen design each router only needs one entry for every other node, drastically reducing memory consumption. This is due to the fact that the number of routers in a network are several orders of magnitude smaller than the number of routes in the Internet.

Another optimization easily implemented due to the split is pre-caching path information. Since paths in the network rarely change, pre-calculating each path allows for a constant lookup time of any pair of nodes. Thus, the path lookup time becomes independent of the network layout. However, pre-caching the path also means handling the annotations along the path correctly. Since this information is possibly updated several times a minute, each link needs to know what path it is related to in the cache. Once this is known, the path cache can be updated as new information is available without having to traverse the entire network.

5.2.2 Query Processing

PaDIS' query processing consists of a Query Manager and the Location Ranker. The Query Manager's job is to interface the information stored in the Network Map Database with the received requests. On the client side, the Query Manager speaks multiple protocols, namely the PaDIS native Protocol as well as the ALTO [14] protocol. The PaDIS native protocol is designed in the spirit of DNS, namely its design goals are connectionless, fast and optimized for machine parsing. PaDIS is aimed at ranking of lists based on network information. As such, it currently has no need to convey ALTO-like information regarding any topology or cost map. Since the ranking of potential sources is the use-case for PaDIS, we only consider this mechanism. Using an ALTO-like approach to communicate whole network and cost maps is out of scope for this thesis. However, the implemented PaDIS prototype does support the basic ALTO protocol.

Furthermore, details of the functions are not revealed to clients and the ranking weights are stripped from the reply. More importantly, no network information is revealed by this scheme, contrary to other schemes of user-provider collaboration [138]. Note, that it is intractable to extract network properties or topologies from the ranking even when combined with traceroute information [1].

Once a request for ranking is received by PaDIS, the Query Manager converts the request into an internal format. This step imperative, since both the PaDIS and ALTO can ask for ranking. After the conversion is done, all available information about each source-destination pair is fetched from the Network Map Database. Since this part of PaDIS is engineered for performance, the path cache significantly improves the lookup times. Each source-destination pair, together with the path information from the database, is then handed to the path ranker individually. The Location Ranker uses a customized ranking function to calculate a preference weight for each path. Once all pairs in the request are augmented with their individual preference weights, the query manager sorts the list by the weights. It then hands a copy of the list off to the Frequent Hitter detection. After that, it strips all additional information added for ranking from the request. Finally, the ordered list is converted back into the protocol on which it was received and sent back to the client that requested the ranking.

The reason for the Location Ranker being separate from the Query Processing is again modularity. In this case, it enables PaDIS to support any kind of customized ranking function. Each ranking function can optimize for different aspects of the path, e. g., network metrics such as bandwidth, hops, delay, etc., results from the Heavy Hitter or a combination of any of these values. Furthermore, it becomes possible for a client to specify their preference regarding the chosen metrics in the query, triggering the usage of an appropriate ranking function. However, the functions are held in the PaDIS server, and clients can only choose between those that are available at the server. In theory, PaDIS' architecture allows for a client to specify custom functions in the request. However, since the client has no other information about the network and does not know how the traffic engineering in the specific network is set up, this has not been implemented. Thus, a client cannot supply its own function to the server but can only use the ones available there.

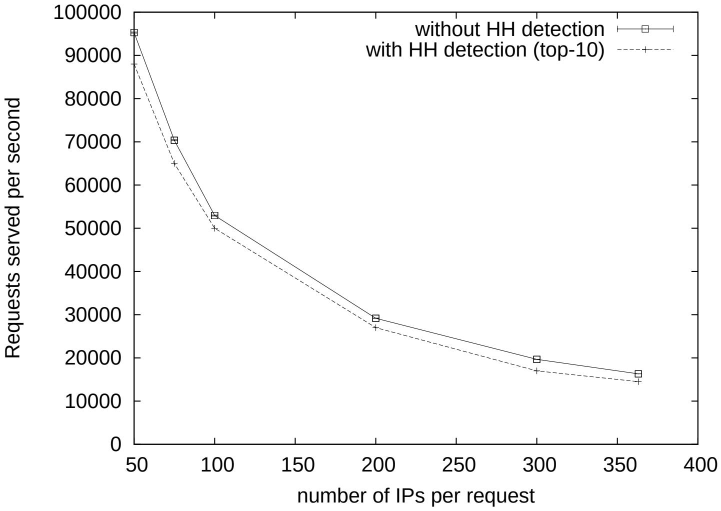


Figure 5.3: **Number of replies from PaDIS with increasing query size**

5.2.3 Scalability and Responsiveness

One of the goals in designing PaDIS is performance. More specifically, the aim is to maximize the throughput in terms of queries per second as well as minimize the processing delay. By decreasing the protocol handling overhead, a first significant performance improvement can be achieved. This is the reason behind choosing UDP as the default transport protocol for PaDIS. Nonetheless, TCP is supported for requests that exceed one MTU, but at a significant disadvantage, as TCP always introduces a connection setup delay that is not required with UDP.

Next we quantify the performance and scalability of PaDIS when using our own UDP-based protocol. For the evaluation, we use two identical dual quad-core CPU machines with sixteen gigabytes of memory directly connected via Gigabit Ethernet. One of the machines is configured to massively issue queries as a PaDIS client while the other is running our prototype implementation of the PaDIS server. In terms of input from the network, we use a static feed of information to the PaDIS server.

To confirm that the topological structure of the network PaDIS is using for ranking decisions does not affect the throughput, we compare the CPU instructions used to answer identical queries when different networks are loaded. In fact, we find that the CPU instructions count per query are identical in every scenario. This confirms that the pre-caching of path information does indeed reduce the complexity to an atomic lookup in the database regardless of the path length and network size.

We perform our throughput evaluation by sending queries to the PaDIS server from a client and measuring the number of received replies. While it is always possible to lose a packet with UDP, our setup with two directly connected machines only

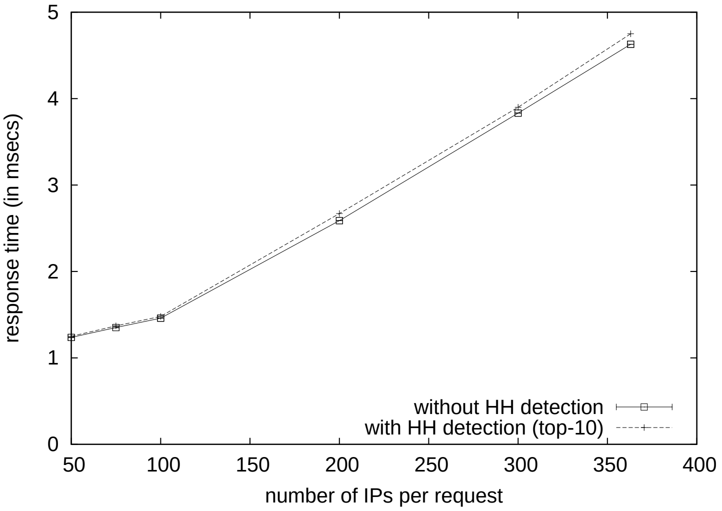


Figure 5.4: **PaDIS' average response time with increasing query size**

drops packets when the buffers of the network cards are full. Thus, an increase in the requests also sees an increase (even if they are not identical) in the replies. As soon as an increase in requests does not trigger an increase in replies, we conclude that the server has reached its maximum throughput. We start at a slow rate and gradually increase the number of requests per second until the PaDIS server is fully utilizing all CPUs and an increase in requests does not result in an increase in replies. Furthermore, we vary the number of possible sources in the requests from 50 (in increments of 50) up to the the maximum number of IP addresses (363) that still fit a UDP-based query. The query is restricted to one MTU, which, in our case was set to the standard 1500 bytes. Also, we run each experiment with and without the Heavy Hitter Detection (HHD) tracking the top 10 ranked IP addresses in each reply. Finally, for each request size as well as with and without the HHD, we repeat the experiment 100 times.

Figure 5.3 shows results of our evaluation in terms of the average number of returned queries per second for the different request sizes. From the results, it becomes clear that the overhead depends on the number of IP addresses (N) embedded in the query. This is no surprise, since the main overhead inside the PaDIS server are the de- and encoding of the packet ($O(N)$) as well as the sorting of the final list ($O(Nlog(N))$. PaDIS is able to serve roughly 90,000 requests per second while ranking 50 IP addresses in our test setup. This number drops to about 15,000 per second when ranking the maximum request size (363 IP addresses). Also, when looking at the performance run with the HHD enabled, it can be seen that the chosen algorithm adds a constant overhead. This is due to the fact that the HHD complexity is given by the number of tracked IP addresses - not by the number of IPs in the requests.

While the replies-per-second can be scaled by making more cores and/or processors available to the PaDIS server, the required processing time per request is fixed by the size of the request. In its current design, each request is processed by one thread, thus negating the possibility to distribute the workload to multiple processors. In order to quantify the delay that occurs when querying a fully loaded PaDIS server, we analyze our test results towards the time that passed between sending the query and receiving the answer.

To our surprise, the response times are much higher than expected from the throughput, see Figure 5.4. When sending the minimum number of IPs, the response time is round 1.2 milliseconds. Since the test setup used two servers that were directly connected, the communication overhead between them can be neglected. This leads to the assumption that the entire time was spent processing the request. However, with 1.2 milliseconds of processing time, and 8 cores to process requests in parallel, the maximum throughput the machine can achieve is around $6,600$ replies per second. In contrast, our earlier measurements showed that the server manages to return $90,000$ requests per second.

The stark difference in the number of replies per second when calculating it from the response time, compared to the measurement comes from buffering. In fact, when rerunning the tests and taking buffering into account, it can be observed that, out of the 1.2 ms, the requests spend more than 1.1 ms in the buffer. In the end, the processing time for a request containing 50 requests is on average 0.08 ms. The same effect can be seen when looking at the results for requests containing more IP addresses. However, as the number of IP addresses increases, the processing overhead also rises. This is due to the fact that sorting the list from the Location Ranker becomes the dominating part of the processing, as it is the only part of PaDIS that does not have a linear complexity.

5.3 PaDIS Usage Options

We now turn our attention to the discussion of possible scenarios of how PaDIS can be used once it is deployed inside a network. First, PaDIS can be used as an ALTO [119] server to offer ISP-aided localization inside an ISP network. The use cases of an ALTO compliant server are manifold [101], e. g., P2P neighbor selection optimization, improved CDI server selection, end-user reachability information, etc. However, offering the full ALTO interface to a third party automatically means sharing potentially private network information with this party. Today, ISPs are reluctant to share this kind of information. Furthermore, by making the network information available in this manner, it becomes possible that different parties are working with different information about the same network. Finally, the possibility to shift preferences depends on the update frequency of the requesting party, not on the changing state of the network. Thus, using the ALTO protocol as the main mode of operation limits the flexibility and responsiveness of the system, while offering the most information about the network to third parties.

For the rest of the evaluation, we shift our focus away from ALTO and focus on the native PaDIS protocol. While this limits the information available to third parties to receiving ranked lists, it also alleviates all of the other concerns of the ALTO protocol. Once an ISP offers the PaDIS service in the same manner as it offers DNS today, it can be used in a multitude of different ways as outlined below:

Client side (a): Clients can install a plug-in to their applications to utilize PaDIS whenever they have a choice between multiple IP addresses. Examples for applications are web browsers optimizing for selection of the nearest CDI server or choosing the best peer in a P2P system.

Client side (b): If applications are unaware or unable to load a PaDIS plug-in, the system library responsible for DNS lookups can be enhanced so that it adds a PaDIS query whenever a DNS server returns more than one IP address (see MultQuery in Section 3.1.2). Furthermore, it is also possible for the client library to collect queries over time and use the aggregated choices (see CrossQuery in Section 3.1.2). However, with this approach it is not possible to augment non DNS-based requests.

Client side (c): Another possibility is to pre-configure the home-router located at the edge of a client's network in the same way as option (b). This is possible thanks to the fact that these routers also often act as DNS-relays. Due to the possibility that more than one machine is using the home router, more DNS queries can be cached at this point, increasing the effectiveness of the CrossQuery selection. Also, along the same lines as (b), it is not possible to act on anything but the DNS requests.

ISP only The ISP can enhance its DNS resolver to contact its PaDIS server and reorder the IP addresses if necessary before returning any answer to a client. Essentially, this is the same approach as before, just at a different scale, since ISP DNS resolvers are used by tens of thousands of users. While this approach has the disadvantage that it manipulates DNS traffic that the ISP is not meant to change, it offers the advantage of being fully transparent to both clients as well as CDIs. The fact that no change is needed at the client side makes this approach appealing to ISPs, since any change and/or deployment of new soft- or hardware to end-users is expensive and cumbersome. Also, ISPs are capable of targeting any number of CDIs.

CDI/ISP Content Delivery Networks can collaborate with ISPs by contacting them before returning their server choices to end-users. However, this adds the additional step of contacting the PaDIS server in their server selection process. In order to identify the best PaDIS server for querying, the CDI needs to contact the ISP where the DNS resolver that sent the DNS query is located.

> This use case has the advantage that content delivery networks can take the ISP preferences into account during their optimization process. Compared to the *ISPs Only* approach, this scheme has the additional advantage that it does not need to change the DNS replies while they are passing through the DNS resolver. It also inherits the property that no change at the client side is needed. On the other side, this scheme cannot target arbitrary CDIs anymore, but needs explicit collaboration with CDIs.

All of the usage options are technically viable. However, when introducing a new service, it is important for fast adoption to keep changes to a minimum. Usually, client-side applications are slow to take off and bring the problem of legacy clients with them. In the end, with client-side changes, a system must always be able to run with and without the service being used. Therefore, all of the client-side options are not practical to be used on the Internet, as it would take too much time, money and effort to deploy them wide enough.

The *ISP Only* scheme brings the advantage of being fully transparent to anyone but the ISP. However, due to its nature of modifying traffic passing through the DNS resolver, we decided to not use this scheme either. This is mainly due to social repercussions that can occur when ISPs start to modify traffic of their end-users.

This makes the *CDI/ISP* scheme the most interesting for further investigation. In fact, since no DNS traffic mangling from the ISP and no client side modification are needed, we are focusing our efforts on this approach.

5.4 CDI/ISP Cooperative Deployment Scheme

Our main architectural motivation when designing the collaboration scheme between ISPs and CDIs is that the server selection is decoupled from the content transfer. Thus, a request for content has in general two phases: The mapping and the transfer phase. In Figure 5.5 we provide a simplified version of how most CDIs handle content requests to their infrastructure. Today, there are two prevalent techniques used to transfer this request: DNS queries and HTTP redirection. We start by illustrating the most commonly used way CDIs map end-users to their servers, i. e., mapping through DNS.

5.4.1 Mapping CDI End-Users via DNS

Each request starts with the mapping phase by the end-user (bottom center in the Figure) issuing a DNS request to the local DNS resolver (1) . Although the full URL of the object is known, only the hostname is resolved here. This means that all information about the content not encoded in the hostname is not available during

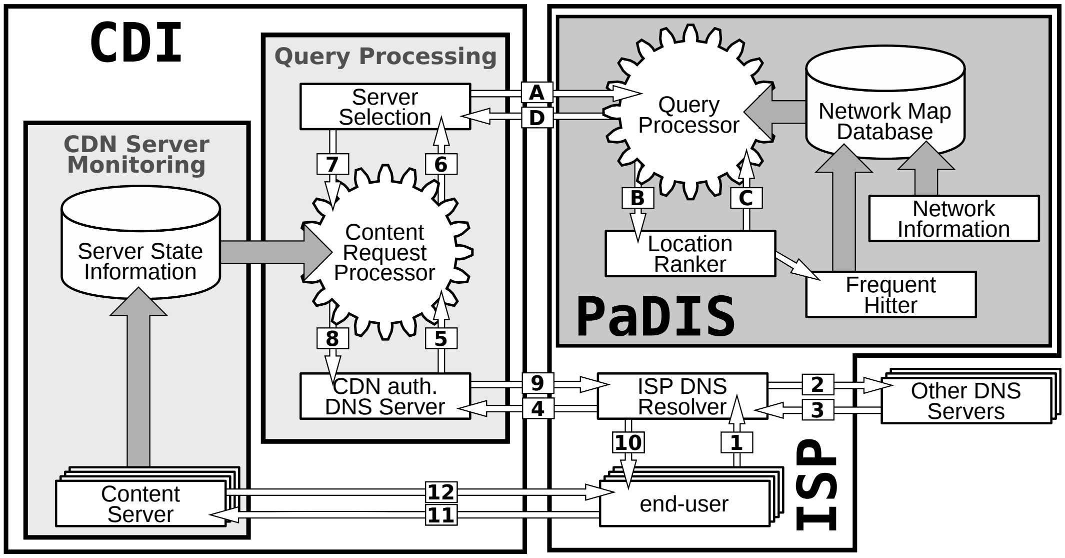


Figure 5.5: **System Setup and interaction when using PaDIS in a collaborative scheme with a CDI employing DNS-based server selection**

the DNS transactions. When the hostname sent by the end-user reaches the ISP-Resolver, it checks its cache to see if an answer to the hostname is already known and still valid. If this is the case, then the DNS resolver returns the answer straight away (10). In the case of the hostname being unknown, iterative requests (2,3) are issued to the DNS hierarchy until the authoritative DNS server for the hostname has been found (4).

The mapping of a CDI is highly complex and unique for each individual CDI rather then being a static mapping of hostnames to IP addresses. But a few generic steps are performed by every CDI. When a request arrives at the authoritative DNS server of the CDI, it hands the request to the Content Request Processor (5). Here, the potential choices that can answer the request are being assembled. Also, current load information for the server and network measurements can be used to further enhance the Server Selection Process. Once this has been done, the request is given to the Server Selection Process (6). Here, a subset of the servers are picked based on a function that best describes the needs of the CDI. Once the subset of servers has been compiled, it is passed back through the Content Request Processor (7) to the authoritative DNS server (8).

Now the normal DNS resolution process continues. With the set of IP addresses available to the authoritative DNS server, it compiles the DNS reply for the end-user and sends it back to the ISP DNS Resolver (9). Here the answer is cached and then forwarded to the end-user (10). Finally, the end-user is in possession of at least one IP address to connect to in order to acquire the content that was requested. With the acquisition of the IP address, the mapping phase ends. Note that no

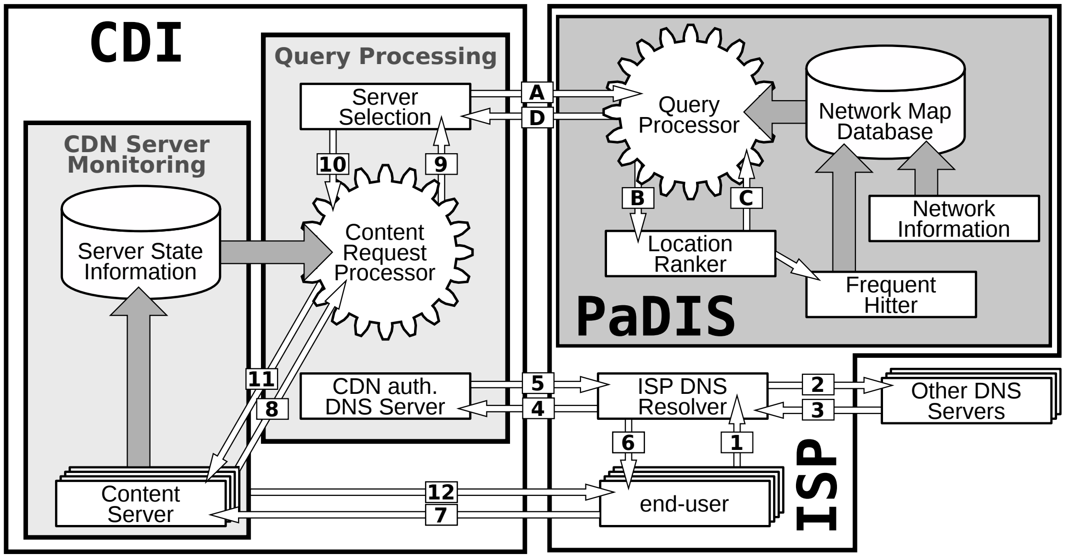


Figure 5.6: **System Setup and interaction when using PaDIS in a collaborative scheme with a CDI employing HTTP based server selection**

content has been transferred so far. All that's left is the content transfer phase. It is straight forward. Since the end-user now knows the IP address of a server, it can simply open a connection there and transfer the content (11,12).

5.4.2 Mapping CDI End-Users via HTTP Redirection

In the case of users being mapped via HTTP redirection instead of DNS, the scheme appears to be almost the same, see Figure 5.6. The DNS resolution process (steps 1 through 6) are the same as with the DNS based server selection, except that the authoritative DNS server is handing back a static assignment of hostname to IP addresses. These usually come with much longer TTLs as well, ensuring that the DNS replies remain cached at the ISP DNS resolver a long time. Putting the DNS resolution process in context with the phases (i. e., mapping and content transfer phase), it becomes clear that it actually belongs to neither.

In fact, when the end-user initiates the connection to the content server (7), it is not clear yet what phase it is in. It depends on the behavior of the content server. If the server is capable of satisfying the request straight away, only static mapping takes place. In this case, the content is directly delivered (12). However, in most cases the first server that an end-user connects to cannot satisfy the content request. In this case, the mapping phase is started. Along the same lines of the DNS mapping, the request is handed off to the Content Request Processor (8), which is responsible for finding the eligible servers that can fulfill the content request. Note that in this case, the object that is being requested is known, since it was transmitted with the original request (7) when the end-user first contacted the content server.

Once the list of eligible content servers has been found, it is handed to the server selection process (9). Here, exactly one server has to be chosen that the user is then mapped to. Once this choice has been made, it is returned to the Content Request Processor (10), which in turn forwards to the content server that received the request in the first place (11). However, at this point the content server is not delivering the content, but instead generates a redirection (usually done by an HTTP 301 code) suppling a new server to the end-user (12). When the end-user receives this redirection page, it opens a new connection to the new content server, which restarts the process at step 1.

5.5 Enabling Cooperation

In both schemes, each request initiates a server selection process. The reason is simple: when operating a large infrastructure with tens of thousands of servers, it becomes necessary to have logic that selects a server for requests. This is also the point where cooperation can be embedded into the process. Firstly, because this step has to be done by any CDIs that relies on dynamic mapping, and is therefore always be available without having to consider the rest of the CDI's technology. Secondly, exactly at this point, the selection process of the CDI knows all eligible content servers and has to reduce them.

When a CDI has compiled a list of eligible servers that both have the requested content and are capable of delivering it from the CDI's point of view, the decision as to what server to chose becomes one of the network. The reason behind this is that the content delivery at this point is determined the CDI's view of the network path between the possible server and the end-user and the inferred network location of the end-user. However, the CDI has merely inferred information about these properties.

Figure 5.5 and Figure 5.6 already contain a simplified version of PaDIS. All parts of PaDIS that are not necessary to answer ranking requests have been aggregated, i. e., the Topology Information, Routing Information, Current Network status and Meta Information have been aggregated into the Network Information. Also, while both schemes differ, the cooperation mechanism through PaDIS hooks into the CDI server selection at the same place. The server list compiled by the CDI has to be reduced to a size that is manageable by the transport protocol as well as the end-user.

Instead of the CDI performing the end-user to server assignment autonomously, we propose that the CDI's Server Selection sends the list of eligible servers and the client IP requesting the content to the PaDIS server together with an objective function (A). There, PaDIS's Query Processor can augment each server-client pair with up-to-date path information. Once that has been done, the requested objective function is applied by the location ranker to each pair, giving it a preference weight in terms of its quality (B). The individual pairs are handed back to the Query processor and are compiled back into a list (C). Once the augmented list has been fully assembled, the Query manager sorts the list by the preference weights in descending order. It then strips the additional information it gathered from the list, compiles an answer packet and sends this back to the CDIs server selection (D). There, the CDI's Server Selection can simply pick the top-most entry in the list, knowing that from a network point of view it is the best choice.

5.6 Effect of PaDIS on Content Delivery

Now that PaDIS has been defined and the schemes CDI/ISP collaboration have been discussed, we turn our attention to an experimental evaluation. To this end, we design, implement and perform a set of active measurements aimed at shedding light on the collaboration between an ISP and CDIs. We start by deploying ten vantage points at residential locations within an ISP. The vantage points all have normal end-user subscription, i. e., DSL connectivity. The downstream bandwidth ranges from 1 Mbps to 25 Mbps while the upstream ranges from 0.128 Mbps to 5 Mbps. Next, we target several CDIs, including the largest CDNs, One-click Hosters (OCH) and Video Streaming Providers (VSP) that are responsible for a significant fraction of HTTP traffic. We rely on an anonymized, passive packet level trace from the ISP to identify the infrastructures using the methodology discussed in Section 3.1. Using active measurements, we are able to extract the diversity they offer in terms of server locations. The measurements started on 1st of May 2010 and lasted for 14 days.

5.6.1 Content Delivery Networks with PaDIS

Using the dataset HTTP-14d from the residential ISP, see Section 3.1.1, we identify the most popular CDI, which is am independent CDN that we refer to as CDN1. Next, we scan the datasets and find more than 3,500 unique IP addresses of servers for CDN1. More than 300 IPs used by servers of CDN1 belong to the same ISP as the measurement points. In addition to standard CDN services, CDN1 also offers an extensive range of service to content producers, including load-balancing and DNS services. When a content producer uses these services, it effectively outsources everything but the creation of content to CDN1.

After augmenting each identified server IP address with its network path information known through PaDIS, we find that the server diversity translates not only into subnet diversity, but also path diversity. Thus, we aggregate the IP addresses of CDN1 into sets that are accessible via the same path. The rationale behind this

object#	01	02	03	04	05	06	07	08	09	10
size	38K	66K	154K	217K	385K	510K	905K	2.48M	6.16M	17M

Table 5.1: **CDN1 performance evaluation: Object sizes**

is that PaDIS is responsible for the path selection, while CDN1 stays responsible for the server selection. After the aggregation has been done, we find 124 different locations with unique paths for CDN1. From each of these locations, we pre-select one server IP address for our measurement. This reduces the number of measured IP addresses to 124 for CDN1.

Each of the ten vantage points ran the measurements targeted at CDN1 24 times a day (once per hour). Each run consists of multiple downloads of pre-identified objects from each of the selected IP addresses. Querying a server for an object without knowing whether the sever is responsible for it is possible because CDN1 makes every CDNized object accessible from any of its servers [66, 129]. Thus, we request the URL directly from each of the pre-selected server IP addresses regardless of the caching strategy of CDN1. Before starting the measurement, we verify that this behavior is true for our selected servers of CDN1. Furthermore, we point out that CDN1 also serves 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. Also, by always asking the same servers, we have a high confidence that the objects stay cached at these IP addresses once they have been primed.

In addition to the static download, we also perform the normal acquisition of the object through the mapping infrastructure of CDN1. This allows for comparing the mapping of CDN1 to the performance of its infrastructure. If the mapping returns an IP address that is already part of the infrastructure we know (i. e., part of the 3500 IP addresses), we use the download that is to be performed for this location instead of downloading the objects an additional time. If the IP address was not known, we downloaded the objects from the proposed server. However, we do not know if the object has already been primed in the cache of the server. Thus, we first prime the cache by requesting the content and then download the object once we know that it is present at this location. This methodology allows us to understand the potential end-user performance improvements. Moreover, we can estimate the network distances and thus the network savings.

The download performance of objects from web pages can depend on the size of the object. Therefore, for CDN1, we select ten different objects with sizes ranging from 36 KB to 17 MB. Table 5.1 gives an overview on the different sizes chosen for the objects. Since we use the raw IP addresses for accessing the various servers and override CDN1s' server selection, we also exclude the domain name resolution time for the CDN1 recommended download.

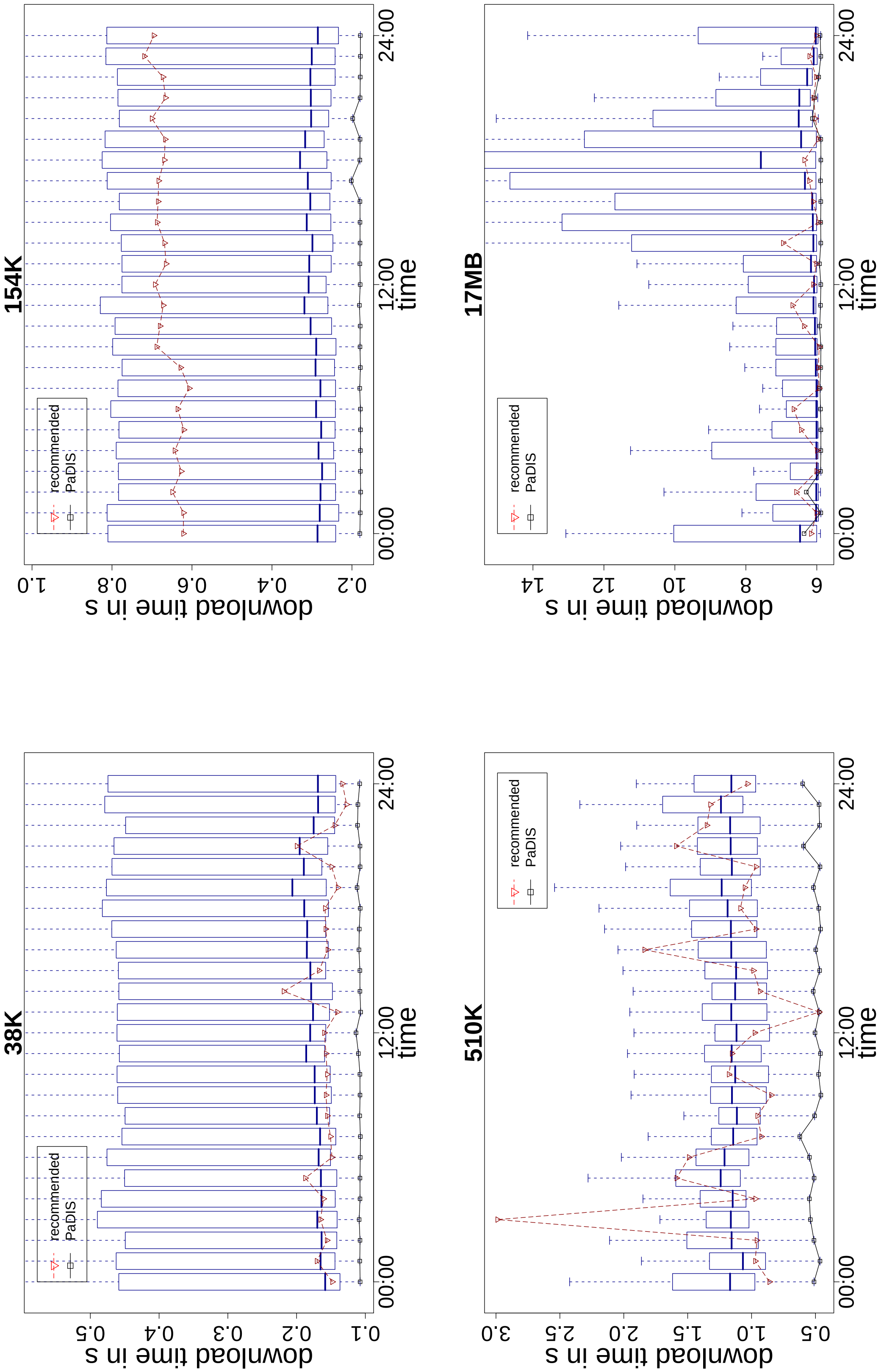


Figure 5.7: **Measurement results for CDN1 on May 12th 2010 for a selection of object sizes**

Figure 5.7 shows a subset of the results obtained from our measurements of one vantage point on a typical day (May 12^{th}). Inspecting the results from the other clients and for the other objects, throughout the experiment we see similar results. 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 object 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 measurement point still within 1.5 times the interquartile range of the lower and upper quartile respectively. The dashed lines with triangles correspond to the object download time for the recommended server of CDN1. The solid line with squares corresponds to the object download time for the server that was chosen by PaDIS based on the current network status while using delay as the function for the location ranker.

Another observation regarding Figure 5.7 is that the download time for the recommended servers of CDN1 are quite good and close to the median download time over all 124 servers. Still, there is significant space for improvement. Firstly, in most cases, 25% of the pre-selected IP addresses give better performance than the server selected by CDN1. When comparing this to the server selection cooperating with PaDIS, it becomes clear that PaDIS manages to use the network status to avoid the network bottlenecks and almost always maps to one of the fastest servers. With respect to benefits for the ISP, we find that PaDIS, by optimizing the traffic for delay, indirectly localizes traffic, i. e., PaDIS chooses closer servers than CDN1 does. In fact, the average path length within the ISP was reduced from 3.8 hops to 3 hops when CDN1 used PaDIS in its mapping of end-users to servers. Overall, through collaboration with PaDIS, CDN1 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 server 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 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.

Next we focus our attention on CDN1's DNS load balancing service. Remember that this service offered by CDN1 only maps users to servers that are run by the content producer. Hence, CDN1's extensive server infrastructure is not used for content delivery in this case. We are interested in finding out if PaDIS can also be helpful in such a scenario. We repeat the experiment of fetching content from the

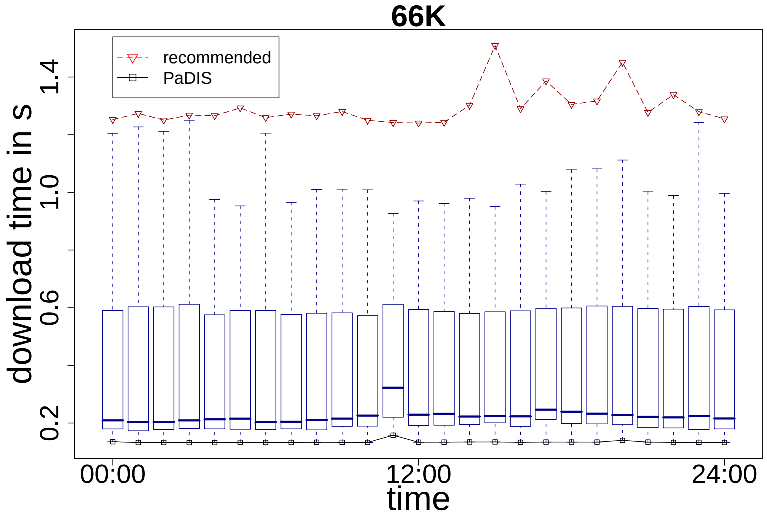
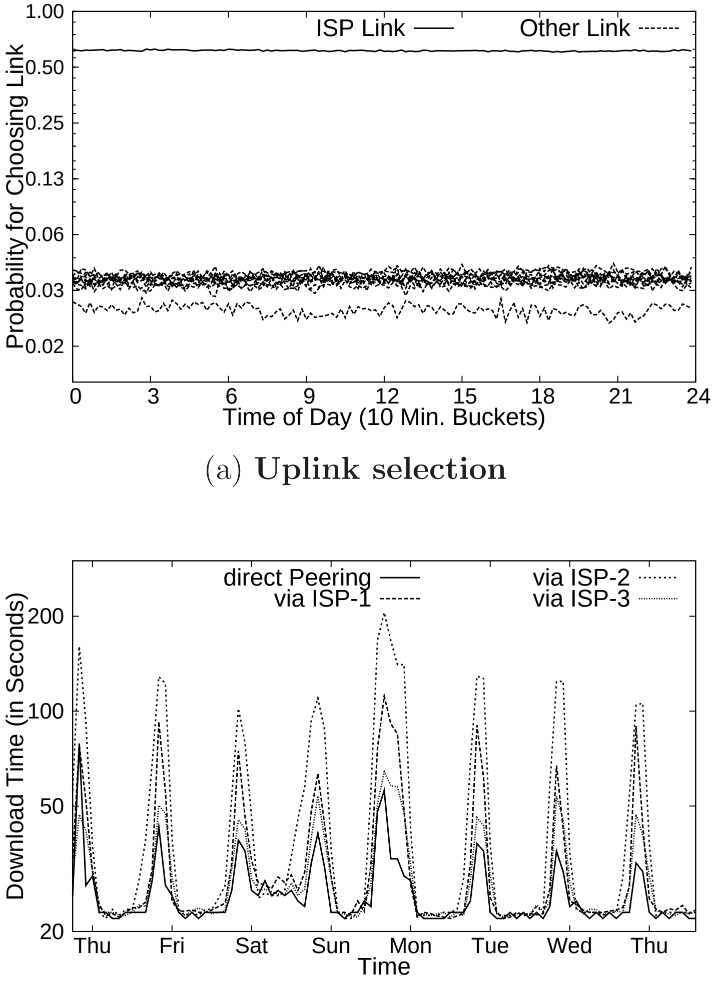


Figure 5.8: **CDN1 DNS load balancer performance evaluation: Boxplot of file downloads across time for object 03.**

CDN1 recommended server as well as from all servers we associated with CDN1. As discussed before, the content of the website is served by all servers from CDN1 even when CDN1 is not responsible for the delivery. However, when looking at the DNS replies from CDN1, we observe that it consistently returns the original server and never any of its own servers.

Figure 5.8 shows the download time of a 66K object (red line) when acquiring the object the standard way, i. e., , through the DNS redirection of CDN1 and then downloaded from the servers of the content producer. At the same time, we repeat the experiment, downloading the same object from all 124 servers with unique paths. The distribution of the download times is again shown as box plots, also in Figure 5.8.

The performance gain between using the content producer's infrastructure to using that of CDN1 is substantial. This plot illustrated very well how CDN1 accelerates content delivery. Almost all of CDN1's servers are faster in the delivery than the content producer's. However, we are fetching the content from servers that are not intended to deliver the content at all. Nevertheless, we were able to use CDN1 infrastructure to improve download times for content that is not supposed to be distributed by CDN1. When using the servers of CDN1 together with PaDIS, we consistently see PaDIS picking the closest server again, confirming our earlier results. We limit the duration of our experiment to one day, as such a behavior is in violation to the agreement between CDN1 and the content producer.



(a) **Uplink selection**

(b) **Download time**

Figure 5.9: **Uplink selection probability and download times via different uplinks for OCH1**

5.6.2 One-Click Hosters with PaDIS

Another class of CDIs are One-Click Hosters (OCH). They differ from CDNs in their server deployment and advertised services. OCHs offer users the ability to share files via a server based infrastructure, typically located within one or several well-provisioned data centers. This means that their approach is more centralized than CDNs while the content they host is produced by end-users and directly uploaded to them. Recent studies have shown that OCHs can achieve better download time than P2P systems (e. g., BitTorrent [15]) for files that are shared between end-users. Therefore, it comes as no surprise that OCHs are overtaking P2P networks in the

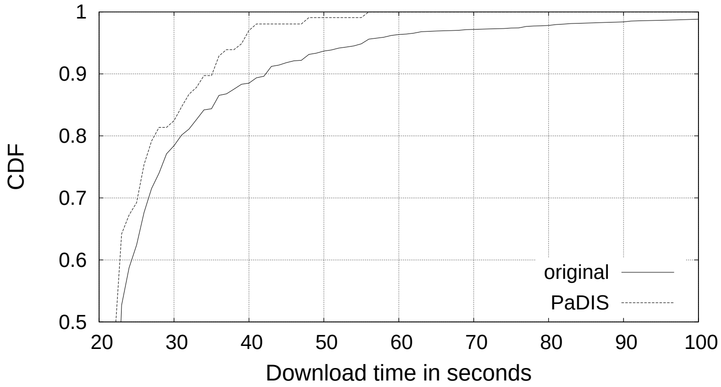


Figure 5.10: **Distribution of download times of a CDI**

role as leading file sharing platforms. Using our data sets from the residential ISP (see Section 3.1.1), we identify the most popular OCH, referred to as OCH1, which, in this data set, 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.

By analysis of our residential ISP traces, uploading a 60 Mbyte test file several times, as well as studying the domain name scheme of the servers, we are able to deduce that OCH1 has twelve uplinks to four different ISPs. The residential ISP we obtain the traces from is among them. To understand how OCH1 does uplink selection, we repeatedly asked the OCH1 for a server to download our 60 Mbyte test for one week starting on April 7^{th} 2010.

First, we are interested in the uplink selection scheme. Since the ISP with our measurement setup has a direct link to OCH1, we expected this link to be chosen unless it is over capacity. The results, in terms of link selection probabilities for the twelve uplinks are shown in Figure 5.9a. Surprisingly, only about 60 % of the requests are directed to a server utilizing the direct uplink to the ISP. Even more surprising is that from the other eleven uplinks, ten are chosen with equal probability while one is chosen with a smaller probability. Furthermore, we observe no time-of-day or time-of-week effects in the link selection for the entire measurement period. In contrast, we find that in our traces the traffic of OCH1 exhibits normal diurnal patterns. This leads us to believe that the link utilization can be improved by using PaDIS' recommendation when assigning clients to servers.

From the mapping replies of the OCH1 link selection experiment, we derive a list of their available servers on a per uplink and ISP basis. Next, we validated that we are able to download our test file from any servers via any uplink. This means, by choosing the server we download our test file from, we also choose the provider uplink that is to be used. This is a prime example of where PaDIS can help content delivery by choosing between servers and, by extension, also choose the path in the network.

To quantify the benefit, - in download time for end-users - that PaDIS can potentially have, we repeatedly download our test file every two hours using each ISP uplink. We run this experiment for one week. Figure 5.9b shows the resulting download times over the course of the week. For presentation purposes and since the performance was very close, we average over all uplinks belonging to one ISP. Our results show that the best download times are possible via the direct uplink, i. e., using the direct peering link. While the download speed during the night is again close to the nominal speed of our measurement probe and does not vary across different ISPs, the download time improvements can be up to a factor of four during peak hours simply by mapping the end-users to the appropriate server with the best uplink.

We finish the analysis of the OCH by looking at the download times from a different point of view. Figure 5.10 shows the distribution of download times when obtaining the test file from OCH1 (original). Furthermore, the curve labeled PaDIS shows the selection that our system would have chosen had it been asked by OCH1. We observe that more than 50% of the downloads do not show a significant difference. These measurement points are taken at night, when congestion is low. However, for 20% of the downloads, we observe a significant difference in the download times. Finally, we drop the maximum download time from over 200 seconds to less than 60 seconds, increasing the download speed by up to a factor of four.

Another point regarding OCH1 is that PaDIS is of limited use with this specific content provider. This is due to the setup of OCH1, i. e., the fastest choice in term of download time is always the direct uplink to the OCH. At no point in time is another uplink significantly faster. Nonetheless, in the case of the direct uplink becoming congested, a selection for the other uplinks is required. In this case, it is beneficial to have PaDIS ready for collaboration.

5.7 Summary

The performance of large scale content delivery by CDIs is hindered by their inability to timely and correctly react to changes in network operation. To overcome this obstacle, we design the ”Provider-aided Distance Information System”, short PaDIS, that acts as an interface between CDIs and ISPs. Its main goal is to supply the missing information, e. g., network state and proximity, to CDIs. Furthermore, we show how PaDIS can be seamlessly integrated into today’s content delivery infrastructure while being fully transparent to any end-user. In fact, only the CDI and the ISP need to be aware of PaDIS. Our prototype implementation of PaDIS shows promising performance results and is able to scale to any network size known today.

With PaDIS fully designed, we run two evaluations regarding PaDIS's effectiveness. First, we design an experiment in which we show the potential of PaDIS if it was used by different CDIs, i. e., a popular CDN, named CDN1, and a large OCH. For CDN1, we focus on the effect PaDIS has on mapping their servers to end-users. Here, we find that, while CDN1 does a decent job in this mapping, PaDIS can significantly improve the performance of the CDN. This is especially desirable for short timescale web transfers, which are mainly delay limited. In addition to CDN1, we also evaluate PaDIS by assuming cooperation with one of the largest OCHs. To this end, we analyze the infrastructure of the OCH and design an experiment that analyzes the potential of PaDIS. Here, we find that PaDIS can significantly reduce the download times from the OCH, which at peak times can be reduced to one fourth.

Thus, we conclude that PaDIS has the potential to substantially improve the end-user experience within an ISP by helping the CDI map its end-users to their servers based on accurate and up-to-date network information.

6
Content-aware Traffic Engineering (CaTE)

So far, the focus has been on improving content delivery for CDIs, and by extension end-users. In this chapter we focus on the other party involved in the collaboration: ISPs. Thus, we introduce the concept of *Content-aware Traffic Engineering (CaTE)* which relies on three key observations. First, a major fraction of the traffic in ISPs is delivered by massively distributed CDI infrastructures (see Section 3.1.3). Therefore, the same content is often available at different network locations with different network paths available to the end-user. Second, the CDI's server selection is decoupled from the content transfer. Thus, it is possible to enhance their server selection strategy (see Section 5.4) in such a way that they become aware of the current network state, the status of links that are traversed and the precise network location of the end-user. And third, PaDIS is available in an ISP network, allowing CDIs to get up-to-date network information to enhance their mapping. With these observations in mind, we design CaTE to improve the network operations of ISPs, the content delivery of CDIs and the perceived end-user performance when downloading content.

6.1 Concept of CaTE

CaTE relies on the observation that by selecting an appropriate server among those being able to satisfy a request, the flow of traffic within the network can be influenced. Figure 6.1 illustrates the concepts by showing how the selection of servers can change the load inside the network and still deliver the same content. We

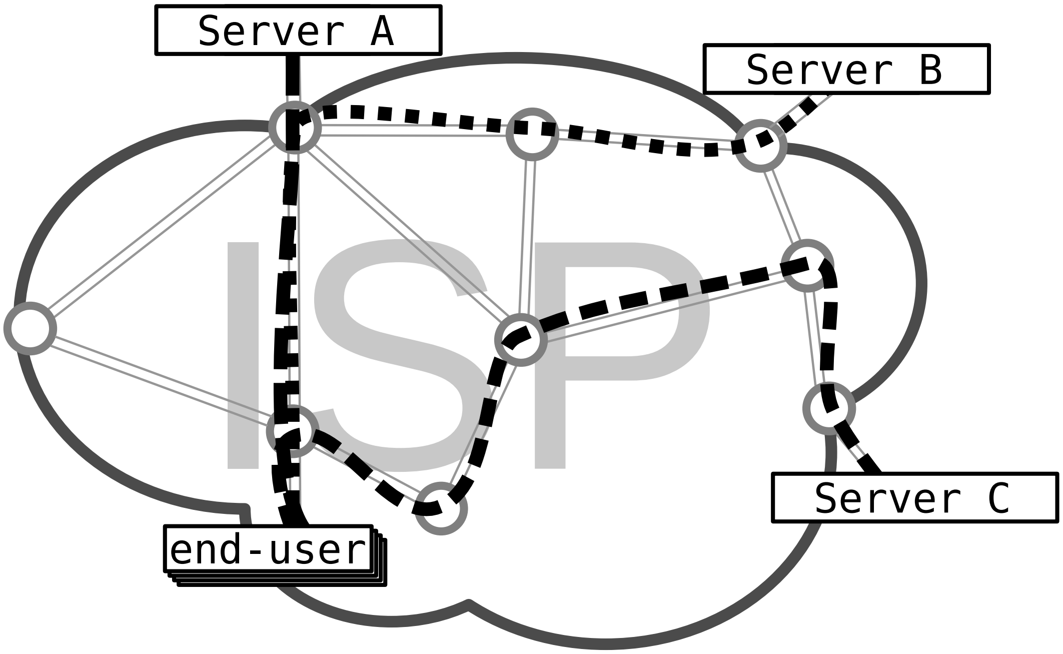


Figure 6.1: **Concept of Traffic Shifting through CaTE by augmenting the CDI Server Selection.**

set all three servers (A,B,C) to offer the same content. When end-users request this content, they get mapped to any of the three servers. However, the path from Server A to the end-users is a subset of the path utilized by server B. Thus, selecting server B will always be worse in terms of network performance when compared to server A. The only reason to choose B over A is when A reaches it capacity to serve the end-user. In contrast, server C has a completely different path in the network. Thus, From a network point of view, it is most beneficial to send most requests to server A (shortest path) and, when this location reaches its limit, to assign the rest of the end-users to server C, since the paths are fully disjointed. Note that this is an example of what the location ranker of PaDIS does. Since custom functions can be used in PaDIS to rank locations, it can propagate any other combination, irrespective of its usefulness.

Without collaboration enabled by PaDIS, CDIs have limited knowledge about the path characteristics inside a network. By allowing CDIs to use PaDIS for alignment of their server selection to the current state of the topology, the network operation can be improved. With better operation of the network, CDI content delivery also improves. In short, given the large fraction of traffic that originates from CDIs and their highly distributed infrastructure, CaTE can shift traffic among paths within a network and, through this, achieve traffic engineering goals for both CDIs and the ISP.

6.2 Modelling CaTE

In order to model CaTE, we first describe how today's traffic engineering works and how CaTE interacts with it. Next, we formalizing CaTE through well known models and finally discuss the relation to multipath routing as well as why CaTE does not induce oscillations.

6.2.1 Traffic Engineering

We start by modeling the network as a directed graph $G(V, E)$ where V is the set of nodes and E is the set of links ($l \in E$). An origin-destination (od) flow f_{od} consists of all traffic originating from the node $o \in V$ (origin) and exiting the network at node $d \in V$ (destination). Note that traffic can enter and exit at the same node ($o = d$). Likewise, the traffic on a link is the superposition of all od flows that traverse the link.

The relationship between link and od flows is expressed by the routing matrix A. The matrix A has size $|E| \times |V|^2$. Each element of matrix A has a boolean value. $A_{f_{od}l} = 1$ if od flow f_{od} traverses link l, and 0 if it does not. The routing matrix A is defined by the internal routing protocol, most commonly OSPF or ISIS. Thus, the configuration of the routing protocol also defines the routing matrix A. Therefore, the traffic on the links induced by the od-flows is directly defined by A. Typically, A is very sparse since each od flow traverses only a very small number of links.

Let $\mathbf{y}$ be a vector of size $|E|$ with traffic counts on links and $\mathbf{x}$ a vector of size $|V|^2$ with traffic counts in od flows, then $\mathbf{y}$=A $\mathbf{x}$. Note, $\mathbf{x}$ is the vector representation of the traffic matrix in a network.

Traditional Traffic Engineering: In its broadest sense, traffic engineering encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic [17]. Traditionally, traffic engineering reduces to controlling and optimizing the routing function and to steering traffic through the network most effectively¿. Translated into the matrix form, traffic engineering is the process of adjusting routing, and through this the matrix A so that the od flow vector $\mathbf{x}$, influences the traffic on the links $\mathbf{y}$ in a desirable way, as coined in [81]. This definition requires that the od flow vector $\mathbf{x}$ is known. For instance, direct observations can be obtained, e. g., with Netflow data [27, 42].

Terminology: We call an od-flow *splittable* if arbitrarily small pieces of the od-flow can be assigned to other od-flows. The assumption that od-flows are splittable is drawn from the fact that each od-flow is the sum of all end-to-end sessions following the same network path. Note that an end-to-end session is *unsplittable*. Since the percentage of traffic of a single end-to-end session is negligible compared to that of an od-flow, we assume od-flows to be arbitrarily splittable.

Let C_l be the set of nominal capacities of the links E in the network G. We denote as *link utilization* ($U_{f_{od}}$) the fraction of the link capacity that is used by od-flows traversing the link. We denote as *flow utilization* ($F_{f_{od}}$) the maximum link utilization among all links that a flow traverses.

6.2.2 Definition of CaTE

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 $\mathbf{y}$.

Not all traffic can be adjusted. Traffic that is reassigned to a different flow must meet the condition of still being able to originate from the new source. When evaluating this in a CDI context with a distributed infrastructure, it means traffic can only be assigned to od-flows that originate from nodes in the network that have at least one sever present able to deliver the requested content. All traffic that is not related to a CDI or does not have more than one eligible origin node cannot be adjusted. We denote the traffic demand that cannot be adjusted as static.

With the knowledge that traffic demands are split into two general classes, i. e., adjustable and static, we rewrite the traffic demand vector $\mathbf{x}$=$\mathbf{x}_s$+$\mathbf{x}_a$ where $\mathbf{x}_a$ denotes adjustable traffic and $\mathbf{x}_s$ denotes the static traffic demands. This changes the way traffic engineering is represented. Thus, we also rewrite the relation between traffic counts on links and traffic counts in od-flows to: $\mathbf{y}$=$A(\mathbf{x}_s$ $+\mathbf{x}_a)$.

Next, we take a closer look at the adjustable demand vector x_a. From the CDI analysis, we know that not all content can be served from all server locations. Furthermore, different CDIs have different server locations. A way to represent the different CDI infrastructures deployments is needed. Thus, x_a has to be further split into vectors describing these properties. In fact, x_a becomes a set of vectors x_{a_i} defined as: $x_{a_i} = p_i t_i$ where t_i is a scalar value of the traffic count and p_i is the *potential vector* describing how the traffic is assigned to the od-flows in x_{a_i}. The potential vector has two types of entries: a) flows that cannot be used since there is no origin server available to deliver the content and b) flows that traffic can be assigned to. Entries of type (a) are 0, while entries of type (b) are the fraction of traffic t_i that is to be assigned to the od-flow. Naturally, a potential vector must distribute all traffic, i. e., the sum of all entries in a potential vector must equal 1.

This changes the traffic engineering further, since the adjustable demand vectors are now the sum of the product of each individual potential vector together with the traffic that can be assigned to it. Finally, let c be the count of different potential vectors that are available. In this case, the full traffic engineering with CaTE is shown in Equation 6.1.

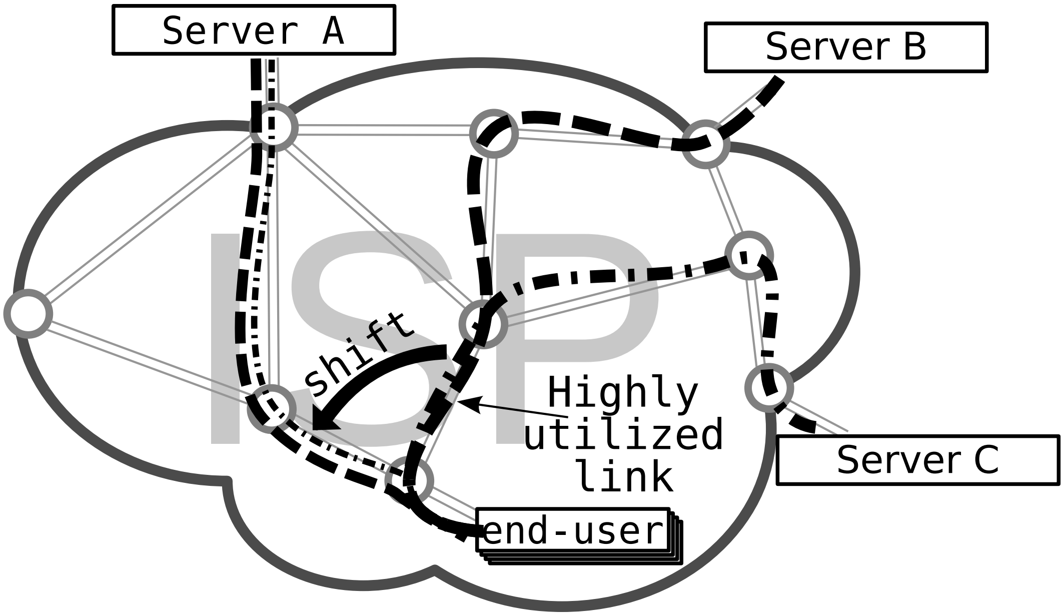


Figure 6.2: **Concept of network path selection through Server Selection**

$$y = A(x_s + \sum_{i=1}^{c} p_i t_i) \tag{6.1}$$

In summary, by adjusting the potential vectors describing the possible sources of content offered by the distributed CDIs, the content demand vector can be changed in a desirable way. This, in turn, changes the network utilization. Thus, applying CaTE means adjusting the content demand to satisfy a traffic engineering goal.

Definition 2: Optimal Traffic Matrix is the new traffic demand vector, $\mathbf{x}^*$, after applying CaTE, given a network topology G, a routing matrix A and an initial traffic demand vector $\mathbf{x}$.

Minimizing the maximum utilization across all links in a network is a popular traffic engineering goal [47, 48, 85]. It potentially improves the quality of experience and postpones the need for capacity increase. CaTE mitigates bottlenecks and minimizes the maximum link utilization by reassigning parts of the traffic traversing heavily loaded paths. Thus it redirects traffic to other, less utilized paths. However, since CaTE is based on the operation and availability of PaDIS in an ISP network, it is easy to see that the traffic engineering goal targeted by CaTE is directly expressed by the function used in PaDIS' location ranker. Thus, by changing the function used by the location ranker, CaTE's traffic engineering can also be changed.

Figure 6.2 again illustrates the CaTE process and focuses on the network utilization when selecting servers. In this case, the path from servers B and C cross an already congested link. With CaTE, the CDI can be made aware of the network bottleneck by advising the CDI to use server A. This results in a better balanced network where the highly utilized link is circumvented. By extension, this also accelerates the content delivery and improves the performance as perceived by end-users.

6.2.3 Interactions of CaTE with Network Operations

To better put CaTE into perspective, in terms of how it fits into the traffic engineering landscape used by ISPs today, we now discuss a few select examples of CaTE's interaction with today's technology. We first briefly outline the interactions with traditional traffic engineering, comment on the similarities with multipath routing and address the question of CaTE inducing oscillations in networks.

6.2.3.1 CaTE and Traditional TE

CaTE is complementary to routing-based traffic engineering because it does not modify the routing and as such does not interfere with the calculations necessary to optimize traffic in a routing based approach. To avoid micro-loops during IGP convergence [49], it is common practice to only adjust a small number of routing weights [48]. 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 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. CaTE complements routing-based traffic engineering and can influence flows at shorter time scales by assigning clients to servers on a per request basis. Thus, CaTE influences the traffic within a network online in a fine-grained fashion.

6.2.3.2 CaTE and Multipath Routing

Multipath routing helps end-hosts to increase and control their upload capacity [74]. It can be used to minimize transit costs [55]. Multipath also enables ASes to dynamically distribute the load inside networks in the presence of volatile and hard to predict traffic demand changes [40, 42, 44, 72]. This is a significant advantage, as routing-based traffic engineering can be too slow to react to phenomena such as flash crowds. Multipath takes advantage of the diversity of paths to better distribute traffic.

CaTE also leverages path diversity to achieve its goals, and can be advantageously combined with multipath to further improve traffic engineering and end-user performance. One of the advantages of CaTE is its limited investments in hardware deployed within an ISP. It can be realized with no change to routers, contrary to some of the previous multipath proposals [40, 44, 72]. The overhead of CaTE is also limited as no state about individual TCP connections needs to be maintained, contrary to multipath [40, 44, 72]. In contrast to [40, 72], CaTE is not restricted to MPLS-like solutions and is easily deployable in today's networks.

6.2.3.3 CaTE and Oscillations

Theoretical results [45, 46] have shown that load balancing algorithms can take advantage of multipath while provably avoiding traffic oscillations. In addition, their convergence is fast. Building on these theoretical results, Fischer et al. proposed REPLEX [44], a dynamic traffic engineering algorithm that exploits the fact that there are multiple paths to a destination. It dynamically changes the traffic load routed on each path. Extensive simulations show that REPLEX leads to fast convergence, without oscillations, even when there is lag between consecutive updates about the state of the network. CaTE is derived from the same principles and thus inherits all the above-mentioned desired properties.

6.3 CaTE Algorithms

We now turn our attention to proposing an algorithm to realize CaTE. A key observation is that CaTE can be reduced to the restricted machine load balancing problem [19] for which optimal online algorithms are available. The benefit of the CaTE online algorithm can be estimated either by reporting results from field tests within an ISP or by using trace-driven simulations. Typically, in operational networks only aggregated monitoring data is available. To estimate the benefit that CaTE offers to an ISP, we present offline algorithms that use traffic demands and server diversity over time extracted from those statistics as input.

6.3.1 Connection to Restricted Machine Load Balancing

Given a set of CDIs and their network location diversity of sources able to serve content, we consider the problem of reassigning the od-flows that correspond to demands of content consumers to the CDIs in such a way that a specific traffic engineering goal is achieved. Given that content requests between end-users and content servers can be re-distributed only to a subset of the network paths described by the potential vectors, we show that the solution of the optimal traffic matrix problem corresponds to solving the *restricted machine load balancing problem* [19]. In the restricted machine load balancing problem, a sequence of tasks arrives, where each task can be executed by a subset of all the available machines. The goal is to assign each task upon arrival to one of the machines that can execute it so that the total load is minimized. Note, contrary to the case of multipath where paths

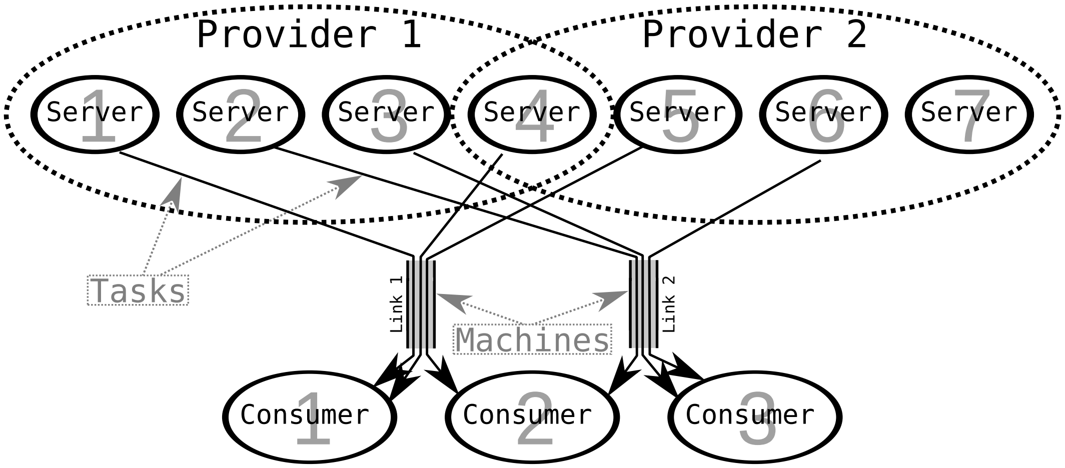


Figure 6.3: **CaTE and Restricted Machine Load Balancing**

between only one source-destination pair are utilized, CaTE can utilize any eligible path between any candidate source and destination of traffic.

For ease of presentation let us assume that the location ranker's function is to minimize the maximum link utilization in the network [47, 48]. Figure 6.3 shows an example of three consumers, where each of them wants to download one unit of content from two different content providers, denoted as provider 1 and provider 2. Given that different servers can deliver the content on behalf of the two providers, the problem consists in assigning consumers to servers in such a way that their demands are satisfied while minimizing the maximum link utilization in the network. Thus, the problem is the restricted machine load balancing, where tasks are the demands satisfied by the servers and machines are the bottleneck links that are traversed when a path, out of all eligible server-consumer paths, is selected. Figure 6.3 shows one of the possible solutions to this problem, where consumer 1 is assigned to servers 1 and 4, consumer 2 to servers 5 and 2, and consumer 3 to servers 3 and 6. Note that the machine load refers to the utilization of the bottleneck links of eligible paths, denoted as link 1 and 2.

To be consistent with our terminology, we define the *restricted flow load balancing problem*. Let J be the set of consumers requesting content in the network, K be the set of content providers that operate the infrastructure, and I be the set of servers for a given content provider, i.e., the set of locations where a request can be satisfied. Note, this set is offered by the CDI in order to satisfy its own objectives and can change over time. We denote as M_{jk} the set of flows that can deliver content for a given content producer k to consumer j.

Definition 3: Restricted Flow Load Balancing Problem is the problem of finding a feasible assignment of flows so that a traffic engineering goal is achieved, given a set of sub-flows $\{f_{ijk}\}$ from all eligible servers $i \in I$ of a given content provider

$k \in K$ to a consumer $j \in J$, and a set of eligible residual flows f_{ij}^{-k}, $i \in M_{jk}$ (after removing the traffic of the above mentioned sub-flows).

Despite some similarities, the nature of our problem differs from the multi-commodity flow and bin packing. In the multi-commodity flow problem [18], the demand between source and destination pairs is given while in our problem, the assignment of demands is part of the solution. In the bin packing problem [29], the objective is to minimize the number of bins, i.e., number of flows in our setting, even if this means deviating from the given traffic engineering goal. Note, in the restricted flow load balancing problem, any eligible path from a candidate source to the destination can be used, contrary to the multipath problem where only equal-cost paths can be used.

6.4 Online Algorithm and Competitiveness

We next turn to the design of online algorithms. It has been shown that in the online restricted machine load balancing problem, the greedy algorithm that schedules a permanent task to an eligible processor having the least load is exactly optimal [19], i. e., it is the best that can be found, achieving a competitive ratio of $\lceil \log_2 n \rceil + 1$, where n is the number of machines. If tasks are splittable, then the greedy algorithm is 1-competitive, i. e., it yields the same performance as an offline optimal algorithm. The greedy algorithm is online, thus it converges to the optimal solution immediately without oscillations.

In the restricted flow load balancing problem, the set M_{jk} can be obtained directly from the potential vectors p_i. The online assignment of users to servers per request, which minimizes the overall load, leads to an optimal assignment of content requests to the eligible flows described by p_i. In our case, flows are splittable since the content corresponding to each content request is negligible compared to the overall traffic in the flows. However, the individual content request is not splittable. Thus, the following online algorithm is optimal:

Algorithm 1. Online Greedy Server Selection. Upon the arrival of a content user request, assign the user to the server that can deliver the content, out of all the servers offered by the CDI, so that the traffic engineering goal is achieved.

6.5 Estimating the Benefit of CaTE with Passive Measurements

Before applying CaTE in real operational networks, it is important to evaluate potential benefits for all of the involved parties. For example, the operator of an ISP network would like to estimate the gains in terms of network performance when applying CaTE. Furthermore, being able to estimate the change in network behavior allows for pre-evaluation of what-if scenarios. Since the behavior of CaTE is directly influenced by the function used in the location ranker, it is beneficial to estimate the changes in network operations when using different functions for optimization. Finally, CDIs also need to be able to to estimate the gains when collaborating with an ISP that is employing CaTE.

In ISP networks, aggregated statistics and passive measurements are collected to support operational decisions as well as routing based traffic engineering. With this information being available, we design and implement a framework that allows for a simulation-driven evaluation of CaTE. To that end, we present offline algorithms that can take passive packet level measurements as input and evaluate the changes in network operations when applying CaTE in different scenarios. We first show a linear programming formulation and then turn to a greedy approximation algorithm to speed-up the process of estimating the gain when using CaTE.

6.5.1 Linear Programming Formulation

To estimate the potential improvement of CaTE we formulate the Restricted Flow Load Balancing problem (see Section 6.3.1) as a Linear Program (LP) with restrictions on the variable values. Variables f_{ijk} correspond to flows that can be influenced. Setting $f_{ijk} = 0$ indicates that consumer j cannot download the content from server i of content provider k. For each consumer j we require that its demand d_{jk} for content provider k is satisfied, i. e., we require $\sum_{i \in M_{jk}} f_{ijk} = d_{jk}$. The utilization on a flow f_{ij} is expressed as $f_{ij} = \sum_k f_{ijk}$.

We use the objective function to encode the traffic engineering goal, i. e., the function used in the location ranker of PaDIS. In this example we again minimize the maximum link utilization across the network. However, any other function based on available network metrics is also possible to use. Let T_e be the set of flows f_{ij} that traverse a link $e \in E$. The link utilization of a link $e \in E$ is expressed as $L_e = \sum_{T_e} f_{ij}$. Let variable L correspond to the maximum link utilization. We use the inequality $\sum_{T_e} f_{ij} \leq L$ for all links. This results in the following LP problem:

$$\begin{aligned}
&\texttt{min}\ L && \\
&\sum_i f_{ijk} = d_{jk}, && \forall\ j \in J,\ k \in K \\
&\sum_{T_e} f_{ijk} \leq L, && \forall\ j \in J,\ i \in I,\ k \in K,\ e \in E \\
&0 \leq f_{ijk} \leq d_{jk}, && \forall\ j \in J,\ i \in M_{jk},\ k \in K \\
&f_{ijk} = 0, && \forall\ j \in J,\ i \notin M_{jk},\ k \in K
\end{aligned}$$

Algorithm 2: Iterative Greedy-Sort-Flow

```
INPUT: $I$, $J$, $K$, $\{f_{ijk}\}$, $\{M_{jk}\}$, $A$.
OUTPUT: $\{f^*_{ijk}\}$.

Initialization:
1. Sort $k \in K$ by decreasing volume: $\sum_i \sum_j f_{ijk}$.
2. Sort $j \in J$ by decreasing volume: $\sum_i f_{ijk}$ for all $k \in K$.

Iteration:
Until no sub-flow is reassigned or the maximum number of
iterations has been reached.
  ▷ Pick unprocessed $k \in K$ in descending order.
    ▷ Pick unprocessed $j \in J$ in descending order.
      ▷ Reassign $f_{ijk}$ in $f_{ij}^{-k}$, $i \in M_{jk}$ s.t. the engineering
        goal is achieved.
```

The solution of the above LP provides a fractional assignment of flows under the assumption that flows are splittable and thus can be solved in polynomial time [75]. The solution is the optimal flow assignment, f^*_{ijk}, that corresponds to the optimal traffic matrix $\mathbf{x}^*$. If flows are not splittable, or the sub-flows are discretized, then the integer programming formulation has to be solved. In this case the Restricted Flow Load Balancing problem is NP-hard and a polynomial time rounding algorithm that approximates the assignment within a factor of 2 exists [86].

6.5.2 Approximation Algorithms

It is common practice to exhaustively study different scenarios to quantify the effect of changes to the network operations. In the context of CaTE, this means changing the traffic demands to influence the traffic matrices which span periods of multiple weeks or months. This amount of data is too large to be solved effectively by algorithms based on a LP approach. Furthermore, each CDI that is added to the list of cooperating parties increases the complexity further, and thus the processing time. To that end, we turn our attention to the design of fast approximation algorithms to estimate the benefits of CaTE in the presence of large datasets. For fast approximations, simple greedy algorithms for load balancing problems [57] are among the best known. Accordingly, we propose a greedy algorithm for our problem which starts with the largest flow first.

Algorithm 3: Greedy-Sort-Flow. Sort CDI infrastructures in decreasing order based on volume they carry and reassign them in this order to any other eligible flow able to deliver the same content which, after assigning the sub-flow f_{ijk} , will yield the desired traffic engineering goal.

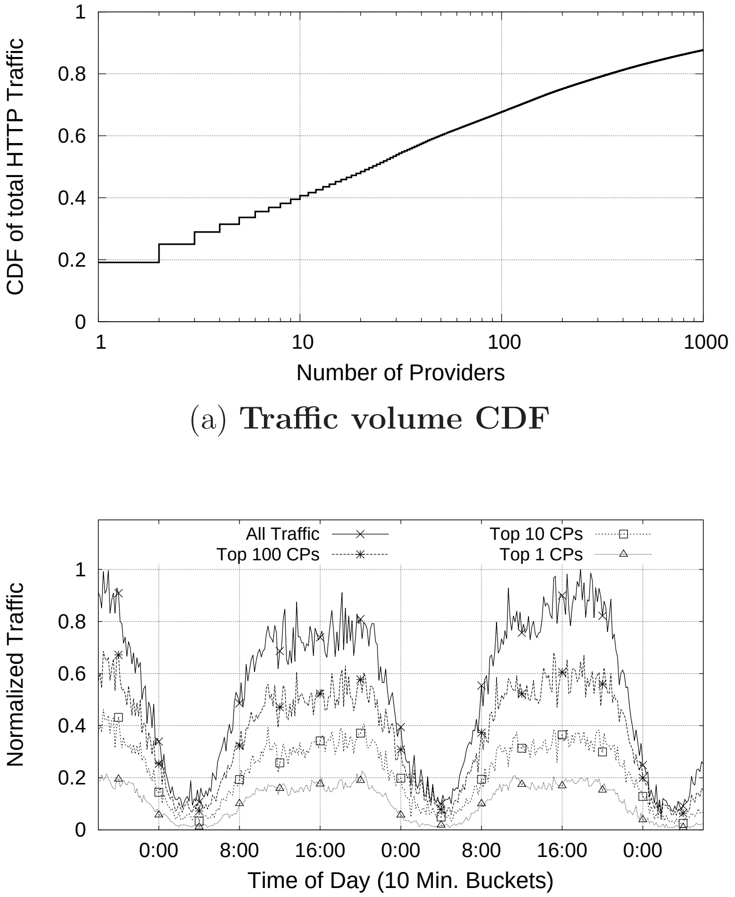
Assignment in sorted order has been shown to significantly improve the approximation ratio and the convergence speed [35, 57]. Recent studies [54, 80, 107] show a large fraction of the traffic originating from a small number of content providers. Therefore we expect the algorithm to yield near optimal results. To further improve the accuracy of the approximation Algorithm 2, we extend it to be *iterative*, by reassigning traffic until no assignment can be found that improves the result. Thus, it also gains the property of converging to the optimal solution. However, our evaluation shows that a small number of iterations, typically one, suffice to provide a stable assignment of flows close to the optimal solution.

6.6 Evaluation of CaTE

Next, we look at quantifying CaTE's potential; specifically, we focus the evaluation towards the benefit potential of the ISP. Again, we use operational data from ISPs, in this case from three different networks. The first network is the same ISP that has been supplying detailed network information and anonymized packet level traces before. The other two ISPs, namely AT&T and Abilene, allow us to evaluate the impact of the ISP network topology.

For a thorough analysis of CaTE, a detailed understanding of the ISP network is necessary, including its topological properties and their implications on the flow of traffic as well as the usage patterns towards different CDIs. This is based on the property that the ISP network influences the availability of disjoint paths, which are key to benefit from the load-balancing ability of CaTE. The usage patterns derive their importance from the amount of traffic that can be attributed to the choices offered by the disjoint paths. Because CaTE influences traffic aggregates inside the ISP network at the granularity of requests directed to the mapping system of CDNs, fine-grained traffic statistics are necessary. Traffic counts per-od flow, often used in the literature, are too coarse an input for CaTE. Thus, a more detailed view on the traffic is needed than traditional traffic engineering offers today.

Earlier studies [54, 80, 107] reported a large fraction of traffic in the Internet is due to a few large CDIs. But an infrastructure operated by a CDI is not necessarily centralized or homogeneous. For example, a large CDI that runs multiple, separated content distribution strategies on subsets of its servers has to be classified properly. We introduce the term *Ubiquitous Content Distribution Infrastructure* (UCDI), which we define as an infrastructure that delivers the same ubiquitously. Note, that with UCDIs, we do not introduce a limit on the size of the infrastructure. This means that a single server hosting a website is classified as an UCDI as well as a globally deployed massive CDI. It can also happen that one CDI is split into several UCDIs, while in the case of a CDI federation, several can be aggregated into a single UCDI. For the identification and classification of the UCDIs, we use the *Infrastructure Redirection Aggregation* methodology described in Section 3.1.3.



(a) **Traffic volume CDF**

(b) **Traffic Evolution over time**

Figure 6.4: **Traffic distribution and normalized traffic evolution over time in ISP1 when split by UCDI**

6.6.1 Data from a Large European ISP

To build the fine-grained traffic demands needed by CaTE, we rely on anonymized packet-level traces of residential DSL connections from a large European Tier-1 ISP, called ISP1. For ISP1, we have the complete annotated router-level topology including the router locations as well as all public and private peerings. ISP1 contains more than 650 routers and 30 peering points all over the world.

While packet level traces for ISP1 already exist, see Section 3.1.1, none span multiple days and also contain HTTP as well as DNS traffic. Thus, we collect a new 10 day long trace starting on May 7, 2010, using the same methodology and setup. We observe 720 million DNS messages and more than 1 billion HTTP requests involving about 1.4 million unique hostnames, representing more than 35 TBytes of data.

In Figure 6.4a, we plot the cumulative fraction of HTTP traffic volume as a function of the UCDIs that we identified by the traffic they originate. Remember, a UCDI is an organizational unit where all servers from the distributed infrastructure serve the same content. We rank the UCDIs by decreasing traffic volume observed in our trace. Note that the x-axis uses a logarithmic scale. The top 10 UCDIs are responsible for around 40% of the HTTP traffic volume and the top 100 UCDIs almost 70% of the HTTP traffic volume. The marginal increase of traffic diminishes when increasing the number of UCDIs. This shows that collaborations with more UCDIs offers a diminishing return as the UCDIs traffic getting smaller. On the flipside, it means that collaboration with a few large UCDIs yields significant potential.

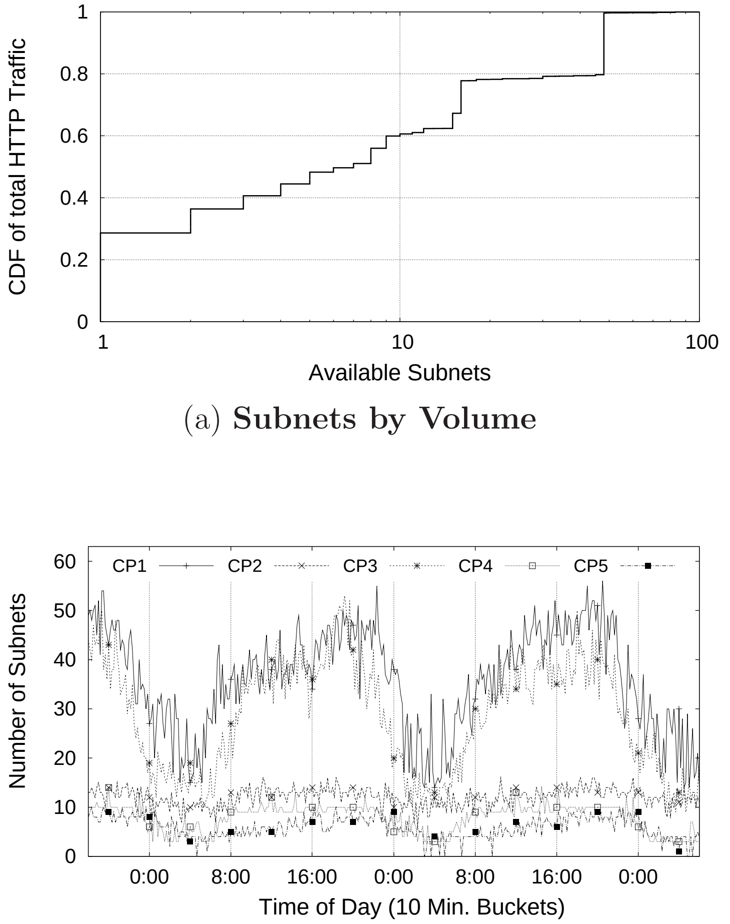
In Figure 6.4b we plot the traffic of the top 1, 10, and 100 UCDIs by volume as well as the total traffic over time normalized to the peak traffic in our dataset. For illustrative purposes, we show the evolution of the traffic. We choose and fix a 60 hour period of our trace, which will be reused for all further analyses as well. A strong diurnal pattern of traffic activity is observed. We again observe that a small number of UCDIs are responsible for about half of the traffic. Similar observations are made for the rest of the trace.

6.6.2 Understanding the Location Diversity of UCDIs

To achieve traffic engineering goals through CaTE, it is crucial to also understand the location diversity of the UCDIs carrying a lot of traffic. This is necessary, as CaTE relies on the observation that the same content is available at multiple locations, and by extension via different, disjunct paths. As routing in the Internet works per prefix, we assume that the granularity of subnets is the finest at which CaTE should engineer the traffic demand. Thus, we differentiate candidate locations of UCDIs by their subnets and quantify the location diversity of UCDIs through the number of subnets from which content can be obtained. Note that the path diversity can be less, as multiple prefixes can follow the same path.

We examine the amount of location diversity offered by UCDIs based on traces from ISP1. To identify the subnets of individual UCDIs, we rely on the internal BGP view of ISP1. Our granularity uses the approach of *Infrastructure Redirection Aggregation* methodology described in Section 3.1.3. Figure 6.5a shows the cumulative fraction of HTTP traffic as a function of the number of subnets (logarithmic scale) from which a content can be obtained. This plot uses the entire 10 days of the trace. We observe that more than 50% of the HTTP traffic can be delivered from at least 8 different subnets, and more than 60% of the HTTP traffic from more than 3 locations.

To finalize our analysis of the location diversity, we turn our attention to the temporal evolution of the location diversity exposed by UCDIs. To this end, we split the trace into 10 minute bins. Figure 6.5b shows the evolution of the number of exposed subnets of five of the top 10 UCDIs by volume. Note that the diversity



(a) **Subnets by Volume**

(b) **Subnet evolution over time**

Figure 6.5: **Subnet diversity analysis and temporal evolution of subnet availability for 5 of the top 10 UCDIs**

exposed by some UCDIs exhibits explicit time of day patterns, while others do not. This can be due to the structural setup or the type of content served by the UCDI. The exposed location diversity patterns, i. e., flat or diurnal, are representative for all UCDIs with a major traffic share in our trace. We conclude that a significant location diversity is exposed by large UCDIs at any point in time, and is quite extensive during peak hours.

6.6.3 Content Demand Generation

The location diversity is not a mere observation about UCDI deployment. It is a direct result of mapping content requests to a server. Nonetheless, it shapes the od-

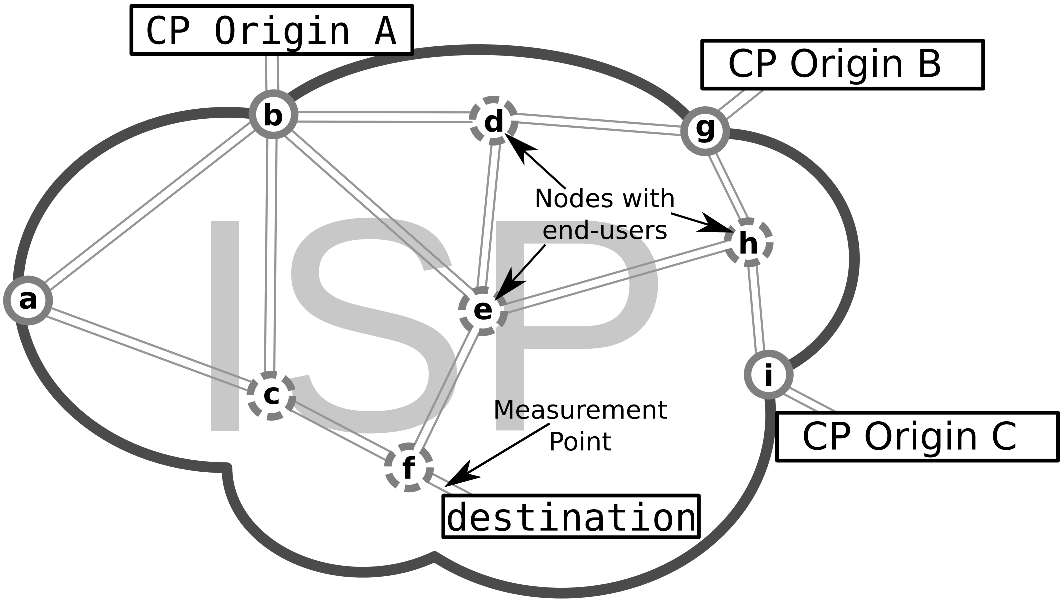


Figure 6.6: **View of the network when generating the potential vectors for an UCDI**

flow based traffic matrix used for traffic engineering, since UCDIs with high location diversity also carry a significant amount of traffic. However, CaTE requires potential vectors that are fine grained on a per UCDI and path basis, see Equation 6.1 in Section 6.2.2. Thus, an od-flow based traffic matrix is not detailed enough. This is mainly due to the impossibility of deriving the potential vectors from the already aggregated traffic matrix. Also, for each UCDI multiple potential vectors can exist. Thus, we propose a solution of how to utilize packet traces and generate a traffic matrix that includes the potential vectors needed by CaTE.

We start by generating the potential vector by bundling and aggregating all requests and responses associated with an UCDI into a set. In detail, we use the IP addresses of the HTTP servers that delivered the content. Also, we use the IP addresses found in the DNS replies as potential servers, even if no HTTP request was made to these IP addresses. In Figure 6.6, this set includes the IP addresses of all origins, i. e., **A**, **B** and **C**. Once the IP set has been created, all origin IP addresses for this UCDI become a potential source. Next, we use the network information from PaDIS to map the sources to the network, translating them into a set of network nodes capable of originating traffic for the UCDI, i. e., the nodes **b**, **g** and **i**. Furthermore, we keep track of the traffic amount that is sent from each origin. Note that each flow originating from these nodes automatically becomes an adjustable entry in the potential vector. In contrast, flows that start at a network node not associated with a node derived from the IP set of the UCDI become a static zero entry in the potential vector, i. e., no traffic can ever originate there.

Next, we turn our attention to the destination of the flows. Our measurement point only gives limited visibility on what is happening in the network. While we know the exact behavior of our users at node **f**, it is currently unknown how the users at the other nodes behave. In oder to approximate the behavior of the overall network, we decide to scale our observations from the single vantage points to the entire network. This puts each node in the role of being a source, a sink or a pure router. An examples for a node that is pure router which only forward traffic is **a**. Likewise, nodes **b**, **g** and **i** are sources for any sink in the network. Furthermore, nodes that connect end-users to the ISP network are are sinks. In Figure 6.6, all sinks are marked with a dotted line (nodes **c**, **d**, **e**, **f** and **h**). Note, that it is possible for a node to be source and sink at the same time. We use the known demand from our measurement and replicate the traffic to the other nodes that have end-users. Next, we use the known traffic volume from the general od-flows and scale the demand for each sink individually in such a way that we reach a link utilization similar to the one already known to us.

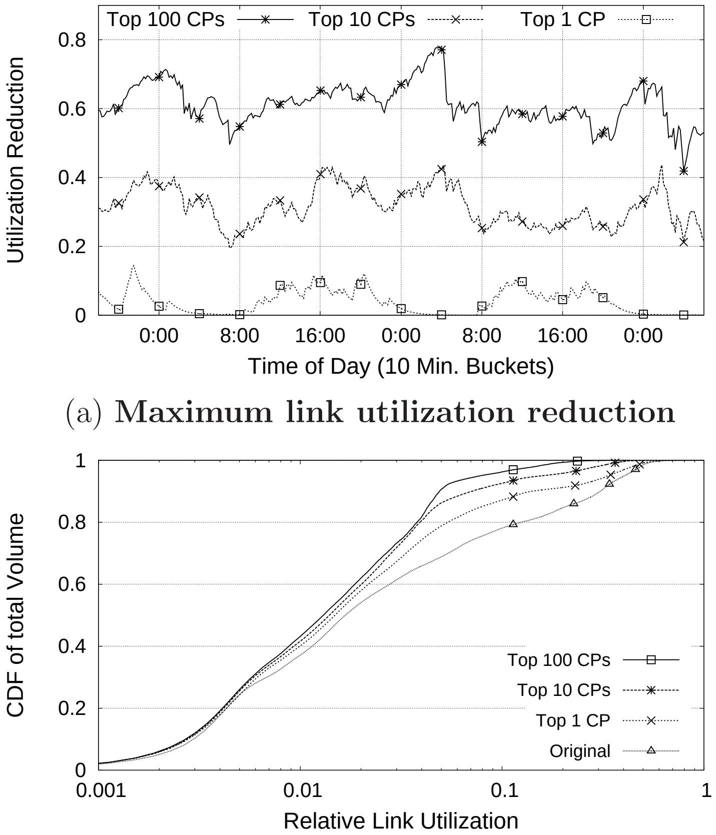
Finally, we are able to generate the individual potential vectors for each sink. For example, the potential vector of the UCDI for sink **f** has three variable od-flows $(b \rightarrow f,\ \ g \rightarrow f\ \text{ and } i \rightarrow f)$ and can freely shift the traffic from between the three paths. This is due to all flows starting at a source that can originate the traffic while ending at a sink that has a demand. In total, the UCDI in this example has five potential vectors due to the five sinks that demand traffic, where each od-flow always starts at a source, and ends at the sink.

We generate the potential vectors for the top 100 UCDIs (by volume) in 10 minute aggregates over the course of the entire 10 days for ISP1 resulting in more than 50 Gbyte of data. For AT&T as well as Abeline, we use the same approach, but generate traffic matrices for only 5 days.

6.6.4 CaTE in ISP1

We start our analysis of CaTE by first considering one of the most popular traffic engineering goals, namely minimizing the maximum utilization of the links in the network [47, 48]. The rationale is that by minimizing the maximum link utilization, network bottlenecks are reduced and the network is used in a balanced way. This, in turn, limits queuing for traffic at routers, what improves the quality of experience. Furthermore, by spreading the load in the network, the ISP is able to postpone the need for increasing the capacity of its links, saving money on infrastructure expansion costs.

With CaTE, an ISP can collaborate with any number of UCDIs concurrently. It is up to the ISP to select the set of UCDIs that are the most important to establish collaboration with. Also, it is possible for an ISP to simply announce its PaDIS to the world, allowing any UCDI to use the ranking service, and through this enable



(a) **Maximum link utilization reduction**

(b) **CDF of Link Utilizations**

Figure 6.7: **Effects on link utilization when using CaTE with the top 1, 10, 100 UCDIs in ISP1**

CaTE's establishment without the need for contracts. When an ISP does not want to publicly announce its PaDIS, it has to consider which UCDIs to cooperate with. Since a significant fraction of the traffic originates from a small number of UCDIs, we consider the most popular UCDIs by volume to evaluate CaTE. By doing this, we focus on prime candidates i. e., UCDIs that a) carry a lot of traffic and b) have significant distribution in their network locations.

First, we focus on the impact of the number of UCDIs that an ISP collaborated with. To this end, we design and perform a sensitivity study to quantify the benefits of CaTE when restricting the UCDIs that can collaborate with the ISP to the top 1, 10 and 100 UCDIs by volume. All traffic of UCDIs that are not eligible for cooperation remains unaffected by CaTE. All traffic that does not belong to the top 100 UCDIs is considered to be static, but is left in the evaluation. Thus, in the case of the top UCDI, only 19% of the traffic can potentially be adjusted, while for the top 10 the fraction increases to 40% and for the top 100 it increases to 70%. For this part of the evaluation, we use the Iterative Greedy-Sort-Flow Algorithm from Section 6.5.2. The function in the location ranker of PaDIS is set to choose the path with the minimal maximum link utilization.

Effect on Maximum Link Utilization. Figure 6.7a shows the reduction of the maximum link utilization over a period of 60 hours when considering the top 1, 10 and 100 UCDIs. We normalized the link utilization by the maximal link utilization when CaTE is not used. The largest gain in maximum link utilization reduction is up to 15%, 40% and 70% respectively. However, while the gain roughly doubles each time more collaboration partners are available, the number of UCDIs cooperating increases by an order of magnitude. This comes as no surprise, since UCDIs added later also contribute less traffic that can be adjusted in a desirable way. Furthermore, those UCDIs no longer have a large infrastructure. This diminishes CaTE's potential to adjust traffic further.

We observe large fluctuations of the reduction of maximum link utilization which are due to the diurnal pattern in traffic (see Figure 6.5a) and location diversity (Figure 6.5b) throughout the day. However, the largest gains are obtained during peak times, when there is more traffic and the highest location diversity is available. Incidentally, this is also when congestion is at its peak and CaTE is most needed. Our results show that CaTE is able to react to diurnal changes in traffic volume and utilizes the available location diversity.

Effect on Distribution of Link Utilization. When traffic is shifted to paths that are less utilized, it is imperative to check how the link utilization in general changes. To make sure that the overall maximum link utilization is lowered, and not just shifted to another link, we analyze our results towards answering this question.

Figure 6.7b shows the CDF of traffic volume in ISP1 across all link utilizations over the entire period of our evaluation. In detail, we take each 10 minute bin and use all link utilizations in it. Since our evaluation spans over 10 days, this results in 1440 bins, each contributing link utilizations for each link in the network. Thus, each link is present 1440 times in the analysis. Again, we normalized the utilization by the maximum link utilization found in the base case, i. e., when CaTE is not used. Furthermore, we consider the scenarios of the ISP cooperating with the top 1, top 10 and top 100 UCDIs.

Turning to the results, we find our expectation of the network utilization being spread in the network confirmed. When considering only the top most UCDI for CaTE, the link utilizations already shift significantly from the highly utilized links to the less utilized ones. Over the entire period of the experiment, the observed maximum link utilization never reaches more than 82% of the the baseline case (e. g., without CaTE). Note that once we drop the dimension of time, these values can be from different buckets. When considering more UCDIs for cooperation with the ISP through CaTE, the link utilizations spread out even more. In fact, when considering the top-10, the maximum relative link utilization is 76% while the evaluation with the top 100 UCDIs shows a drop in maximum link utilization to 72%.

This confirms our expectation that CaTE shifts the traffic away from highly utilized links to those less utilized, and by extension, spreads the load in the network.

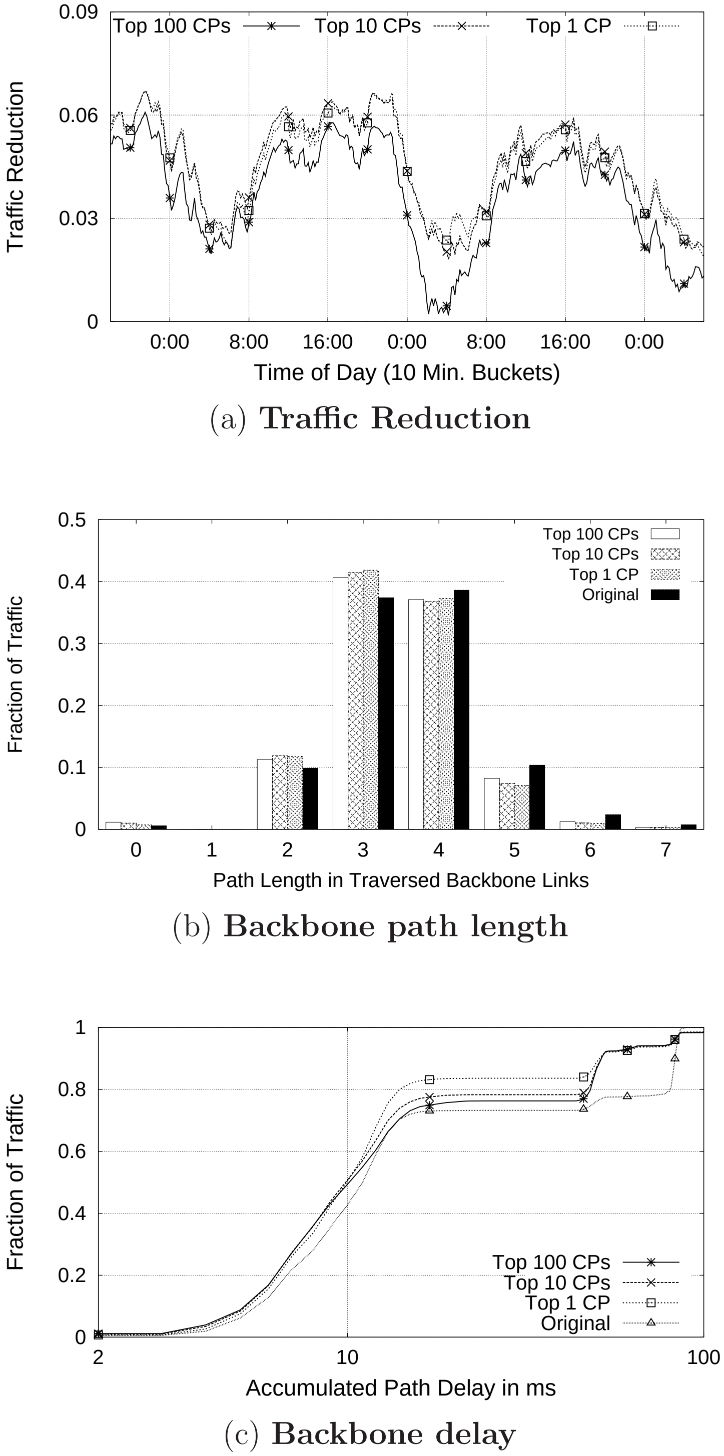
Effect on Network-wide Traffic. Although the location ranker function is set to optimize for link utilization, CaTE manages to reduce the overall traffic in the network, e. g., the sum of all traffic on all links, see Figure 6.8a. This is due to a tie-break decision in the ranking function. In the case of multiple paths being equal in their utilization, the tie is broken by choosing the shorter path in terms of hops. This enables the side effect of content being fetched from closer locations, which in turn uses less links and thus reduces the overall traffic in the network.

Quantifying the results obtained with CaTE, we find network-wide traffic reduction of up to 6%. Also, the reductions follow a clear diurnal pattern, with the reductions being significantly higher when they are needed most, i. e., during peak hours. Furthermore, when considering the top 10 UCDIs, the total traffic reduction is very close to that while considering the top 100 UCDIs, indicating that the gains in reduction of network-wide traffic are almost negligible beyond a certain number of considered UCDIs. From an ISP's perspective, CaTE reduces the overall traffic inside its network, making it more competitive, as it can serve more end-users with the same infrastructure. Additionally, it can delay investments in capacity upgrades and improve end-user satisfaction through a higher quality Internet experience.

Effect on Traffic Path Length. Our results show a reduction of the overall traffic in ISP1, which we attribute to an overall reduction of path length. In order to confirm that the traffic reduction is indeed caused by a drop in path length, we quantify the change path length inside ISP1 in Figure 6.8b. For each path hop count in the ISP network, we show the fraction of traffic that is being traversed by it. The bars labeled *Original* show the base case when CaTE is not being used. Likewise, the bars labeled top 1, top 10 and top 100 show the fraction of traffic on the given path length when UCDIS are cooperating with the ISP through CaTE.

We find that CaTE indeed shifts traffic to shorter paths. In the case of this ISP, the shift can be most prominently seen between the path length 2 and 3. Here, the fraction of traffic rises in all the considered cases, meaning that traffic is shifted onto paths with this length. At the same time, the path length 4 and 5 considerably drop in their fractions of traffic. Thus, we confirm the redirection of traffic to paths with the same or shorter length than the ones used without CaTE. However, in rare cases we also notice that CaTE shifts traffic onto longer, but very lightly utilized paths.

Note that there is almost no traffic on path length 0 and 1. This is due to the network design of ISP1. Also, the plot is cut off at path length 7, since the traffic on longer paths is negligible. We conclude that applying CaTE to a small number of UCDIs yields significant improvements in terms of path length reduction for the majority of the traffic.



(a) **Traffic Reduction**

(b) **Backbone path length**

(c) **Backbone delay**

Figure 6.8: **Effect of CaTE on secondary network performance parameters when cooperating with the top 1, 10, 100 UCDIs**

Effect on Path Delay. Finally we turn our attention to the effect on the delay for traffic. Note that the objective is still to minimize the maximum link utilization. Thus, similar to the traffic reductions, CaTE does not directly optimize towards reducing the path delay. However, by reducing the utilization as well as the traffic and the path length, we also expect a lower path delay to be achieved by CaTE.

We add the the path delay to our simulation through static values known from the network operation of ISP1. Furthermore, we do not apply any model to increase the delay when links become utilized, but keep it at the static value regardless of the utilization. Finally, the simulation ignores all delay that comes from non-backbone links. This includes the entire aggregation network as well as potential delay from peer or links in the networks.

Figure 6.8c shows the accumulated path delay for the traffic that flows within ISP1. We find that for the majority of the traffic, the path delay does not change significantly. However, for the part of the traffic that traverses long backbone links, significant improvements can be observed. While CaTE is not able to eradicate all long delay, it significantly reduces backbone path delays over 50ms.

The numbers for the backbone path delay are relatively modest compared to the values for the access part of the network [90]. However, improving the access delay requires significant investments as it can be done mostly through changes in the access technology, e. g., from copper to fiber. When considering the end-to-end delay, the delay of the path outside the ISP's network also needs to be considered. As content infrastructures are located close to peering points [7, 79, 80], e. g., IXPs or private peerings, delays are expected to be relatively small, especially for popular UCDIs. Estimating the impact CaTE has on end-to-end performance for every application is very challenging, due to the many factors that influence flow performance, especially network bottlenecks outside the considered ISP.

Summary. Our evaluation shows that CaTE yields encouraging results, even when only a few large UCDIs are collaborating with an ISP. In fact, even metrics that are not directly related to the optimization function of CaTE are improved. Besides significant improvements for the operation of ISP networks, the end-users are expected to also benefit from these gains. This can be attributed to the decrease of delay as well as the reduced link utilization.

6.6.5 CaTE with other Network Metrics

So far we have evaluated CaTE with one traffic engineering objective, namely, the minimization of maximum link utilization. However, CaTE is based on PaDIS, which allows for customized functions in the location ranker. Thus, ISPs and UCDIs can optimize for other network metrics, i. e., path length or path delay. To this end, we complement our evaluation by evaluating CaTE when using path length and delay and compare it with the results from the previous analysis (Section 6.6.4). We limit the presented results to the case of CaTE collaborating with the 10 UCDIs, due to the observation that most of the benefits CaTE brings can be achieved when cooperating with a small number of large UCDIs. Furthermore, we check that similar observations are made when applying CaTE to the top 1 and 100 UCDIs.

In Figure 6.9a we plot the total traffic reduction when applying CaTE to the top 10 UCDIs with different optimization goals. The first observation is that when the network metric is path length, the total traffic reduction is the highest, with up to 15%. The total traffic reduction when optimizing for path length is as close to the one achieved when the metric is delay. Optimizing for other metrics provides the expected result: the optimized metric is significantly improved, but at the cost of not optimizing other metrics as much. For example, optimizing for link utilization diminishes the benefits from path length (Figure 6.9a) and vice-versa (Figure 6.9b). Still, significant improvements can be achieved even when optimizing for another network metric and we encountered no case of significant deterioration in any of the network metrics throughout our experiments, see Figure 6.9. Also, the question of how the different metrics interact when being mixed with CaTE is an interesting topic, and is part of planned future work.

6.6.6 CaTE in AT&T and Abilene

To quantify the potential benefits of CaTE in networks with different topological structures than ISP1, we repeat our experiments for two more networks: AT&T and Abilene.

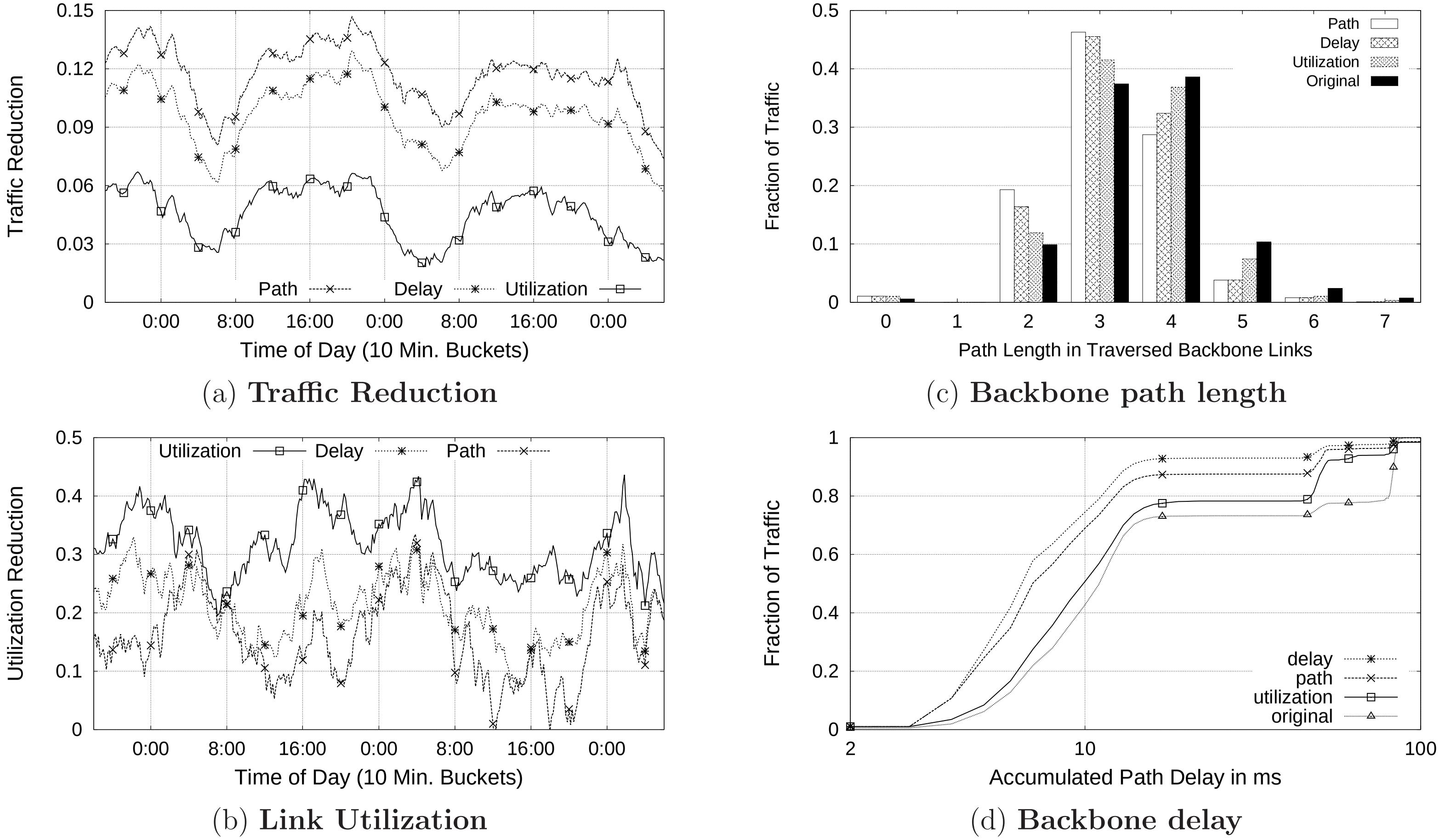
AT&T is one of the largest commercial networks. We use the topology for the US backbone of AT&T with 113 nodes as measured by the Rocketfuel project [121, 125]. Given that no publicly available traffic demands exist for AT&T, we rely on the gravity model [112] to generate several traffic demand matrices as in ISP1.

Abilene is the academic network in the US. We use the Abilene topology and traffic demands covering a 6 month period that are both publicly available.[1]

The topologies of both networks differ significantly from the one of ISP1. In AT&T, many smaller nodes within a geographical area are aggregated into a larger one. Abilene has few but large and well connected nodes with a high degree of peerings. For the application mix we rely on recent measurements in AT&T [54] and for server diversity we rely on measurements of users in these networks [7]. Finally, we assemble the potential vectors needed for CaTE by basing the request pattern on the one from ISP1, as no traffic traces from these networks are publicly available.

Figure 6.10 shows the cumulative fraction of normalized link utilizations for AT&T and Abilene with different optimization goals. We again only consider the top 10

[1] http://userweb.cs.utexas.edu/~yzhang/research/AbileneTM/



(a) **Traffic Reduction**

(c) **Backbone path length**

(b) **Link Utilization**

(d) **Backbone delay**

Figure 6.9: **Effect of CaTE on network performance metrics when using different optimization functions**

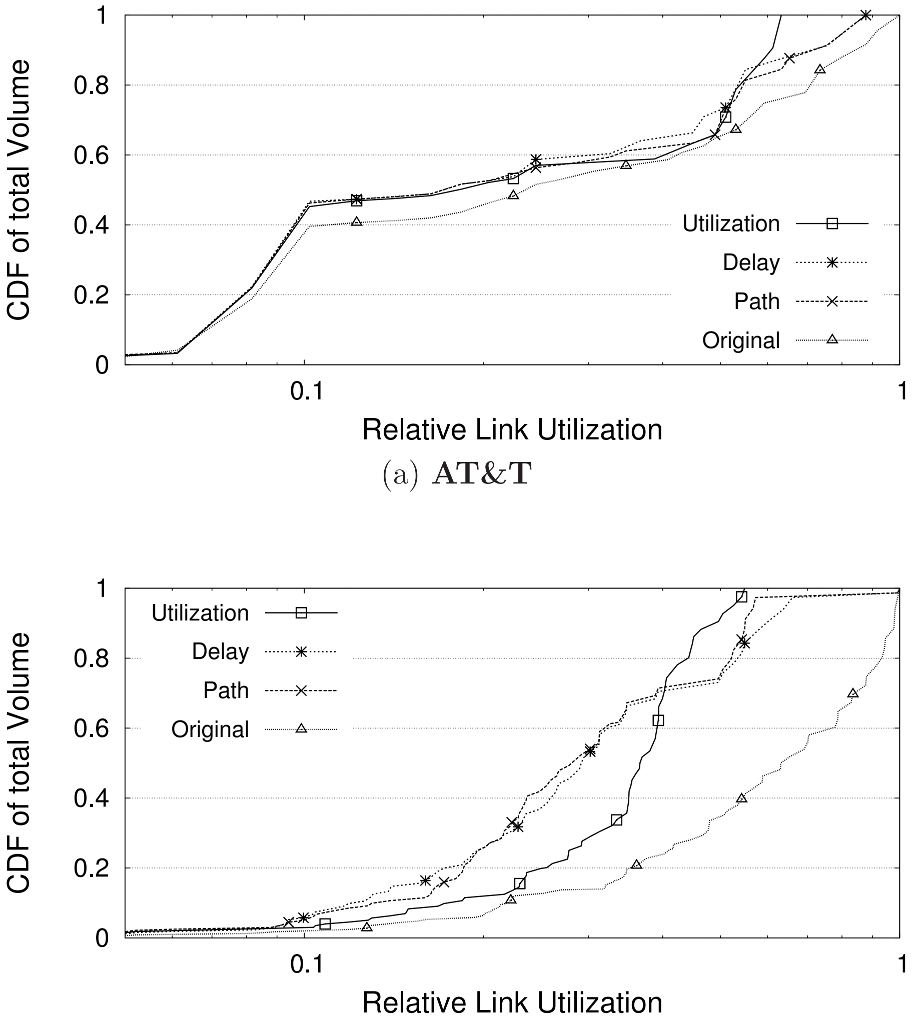
UCDIs for cooperation through CaTE. Along the same lines, all traffic that is not eligible for CaTE remains static and unaffected.

For AT&T, the benefit for the maximum link utilization is about 36% when the network is optimized for minimizing the maximum link utilization, while the median reduction in terms of network-wide traffic is about 3.7%. When other optimizations are used, the benefits of CaTE regarding the link utilization minimization are approximately 12% for path length and delay. However, when looking at the median traffic reduction of these metrics, the traffic is reduced by 5.4% when path length is used, while delay achieves a reduction of 5%. In the Abilene network benefits of CaTE are more significant: 45% reduction in the maximum link utilization and 18% for network-wide traffic when CaTE optimizes for link utilization. When targeting the other two metrics, i. e., path length and delay, the results show that CaTE does not reduce the maximum link utilization. In fact, the maximum link utilizations stays constant. This is due to the structure of the network and the fact that the content is available closer, but at the cost of keeping the high utilization on some of the links. However, when looking at the median traffic reduction, both metrics manage to reduce the traffic by over 24%. These results show that CaTE is capable of targeting different optimization goals in different network structures and is able to optimize for different metrics.

It is worth noting that for AT&T 40% of the links have a normalized link utilization less than 10% while the remaining link utilizations are distributed almost linear. This distribution fits the structural observations made for the AT&T network: many links from smaller nodes are aggregated into larger ones. This also explains why the benefits for AT&T are smaller, since such a structure reduces the path diversity. Turning our attention to Abilene, we attribute the higher reduction of maximum link utilization and network-wide traffic to the non-hierarchical structure of the network and a higher ratio of peering locations. Applying CaTE to both AT&T and Abilene networks where the network metric is delay or path length shows similar behavior in CaTE as it does in ISP1.

6.6.7 Case Study: Netflix in ISP1

We finalize our evaluation of CaTE by evaluating a potential future scenario that ISP1 might be faced with. Today, the launch of new content hosted on UCDIs such as high definition video or others that share flash-crowd characteristics, is not done in coordination with ISPs. This is challenging to ISPs that have to deal with rapid shifts and increase of traffic volume as currently deployed traffic engineering tools are too slow to react to these rapid changes. Furthermore, the end-user experience for popular applications is far from optimal as application designers have limited means to optimize the end-to-end delivery of content [79]. Thus, both ISPs and applications would benefit from the improved traffic engineering capabilities offered



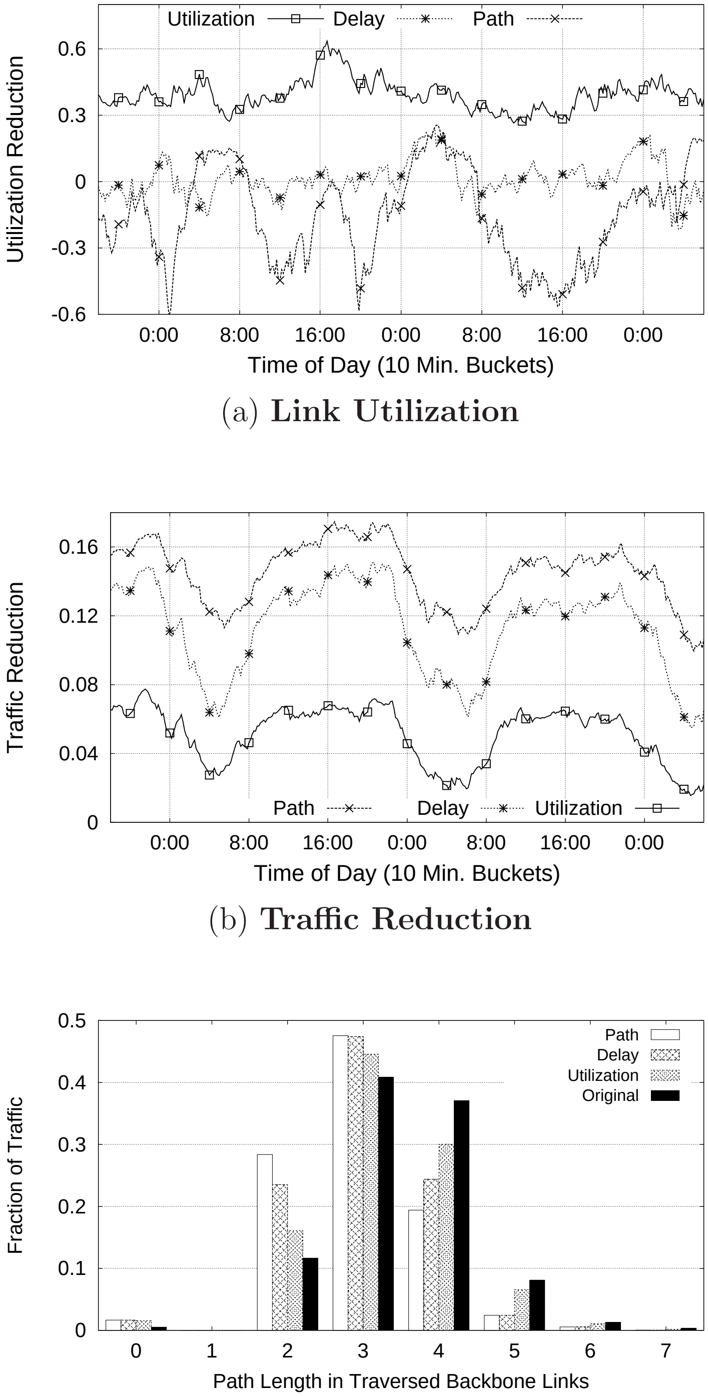
(a) **AT&T**

(b) **Abeline**

Figure 6.10: **Link utilizations with CaTE in AT&T and Abilene**

by CaTE even when planning the launch of an application. We believe that CaTE can also be used as a planning tool for future network deployment.

We choose the case of Netflix, a very popular application that delivers high quality videos over the Internet. It relies on commercial CDNs such as Level3 and Limelight for its content delivery. Today, Netflix is available only in North and Latin America, and is announced to arrive in the UK soon. Recent studies show that Netflix is responsible for more than 30% of the peak downstream traffic in large ISPs [114] in North America. Thus, we consider the scenario of Netflix launching its service in the large European ISP1. If the launch happens overnight, ISP1 has to deal with a huge amount of highly variable traffic that grows rapidly. This has significant implications



(a) **Link Utilization**

(b) **Traffic Reduction**

(c) **Backbone path length**

Figure 6.11: **Effect of CaTE on network performance metrics in the scenario of Netflix deployment in ISP1**

on the operation of ISP1. With CaTE, the traffic generated by Netflix can be better spread in the network, as shown in our extensive evaluation with the current content demands. Thus, CaTE can potentially turn the negative consequences imposed by additional traffic into an opportunity for ISP1 by delivering Netflix reliably and with limited negative network effects deteriorating end-user experience.

To quantify the effect of Netflix being deployed in ISP1, we simulate the launch by assuming that the UCDI currently hosting Netflix increases its traffic 20-fold, while keeping the distribution of the requests. Next, we generate a new set of traffic demands for CaTE accordingly. Note that the potential vectors for CaTE stay constant as the infrastructures already exists and does not get extended in this evaluation. After the traffic increase has been applied, we again consider the top 10 UCDIs by volume for CaTE, and show the benefits when optimizing for different metrics.

Our results show that with CaTE, the load of the most utilized link can be reduced by up to 60% (see Figure 6.11a), the total HTTP traffic volume can be reduced by 15% (of Figure 6.11b) and traffic can be shifted towards shorter paths inside the network of ISP1 (Figure 6.11c). The increased gains that CaTE can offer compared to the traffic today stem from the fact that the infrastructures being used by Netflix are highly distributed. Thus, more traffic can be shifted between the already substantial server, and by extension, path diversity.

However, when considering all metrics, we again observe that they cannot all be optimized to their full extent at the same time. For example, a reduction of traffic in 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 minimizing the maximum link utilization is chosen as the optimization function for CaTE, benefits in all metrics can be observed.

Internet applications such as Netflix are in a position to negotiate their deployment in order to improve end-user experience and not disturb the operation of ISPs. CaTE can be used to identify the best peering points between the UCDIs that deliver Netflix traffic and the ISPs that receive its traffic. In addition, ISPs might offer better peering prices if the UCDIs hosting Netflix are willing to provide a higher diversity in the locations from which the traffic can be obtained. This would lead to a win-win situation where Netflix can offer a better service to its users, the UCDIs achieve reduced pricing on their peering agreements, and ISPs can compensate the reduced peering revenue through more efficient operations.

6.7 Summary

PaDIS needs to have a positive effect on CDI operation as well as that of ISPs in order to be adopted quickly and broadly. Thus, we introduce and evaluate the concept of Content-aware Traffic Engineering (CaTE) as the counterpart to improving the CDI performance through PaDIS. In fact, CaTE is based on three key requirements. First, today's CDIs are responsible for a major portion of Internet traffic. Second, CDIs use a massively distributed infrastructure that allows them to deliver content from a multitude of locations. And third, PaDIS is able to aid CDIs in their server selection by using up-to-date network information.

It is the third requirement that enables CaTE to also improve the operation of ISPs when collaborating with CDIs through PaDIS. By giving ISPs the ability to advantageously influence the server selection of CDNs, an ISP can choose the path traffic takes through its network. This ability leads to the effect that an ISP can perform traffic engineering at much shorter timescales as well as being able to target specific optimization metrics.

In our evaluation of CaTE, we find that it not only improves the operation of CDIs, but also significantly enhances an ISP's network management. In fact, we show that for a residential ISP, CaTE reduces the maximum link utilization during peak hours by up to 40% while reducing traffic by almost 6% when collaborating with only 10 Content Providers. Furthermore, CaTE is able to reduce the delay between end-users and content servers, which confirms our results from the evaluation of PaDIS (see Section 5.6). Finally, we show that the benefits CaTE brings are not an artifact of the ISP we evaluate by adding two more networks to our evaluation, namely Rocketfuel AT&T and Abeline. In both networks we confirm the trend that CaTE is able to increase the efficiency of content delivery while reducing the load on the networks.

CaTE is a new tool in an ISP's arsenal for traffic engineering. It is designed as a complementary tool that enables an ISP to do fast paced, near-realtime traffic engineering. As such, CaTE gives ISPs the ability to turn the volatile traffic shift induced by CDIs into an opportunity to perform fast-paced traffic engineering.

7 Future Work

PaDIS and CaTE are designed to enable cooperation between CDI and ISPs by making one existing infrastructures operations aware of the other. But cooperation does not need to stop at this point. In fact, the deployment and acceptance of cloud services opens new possibilities increasing the effectiveness of PaDIS and CaTE even further. We base this on the fact that ISPs have been moving towards deploying general-purpose computing and storage infrastructures in their points of presences (PoPs). While small, specialized hardware deployments were always needed by ISPs (DNS servers, access aggregation, etc) in their PoPs, the needed infrastructures in order to deploy new, competitive services is made possible by the convergence of computing, storage, communications of cloud services. Examples of these services are internet based telephone and video communication systems, video on demand portals and old-fashioned television via Internet. Most of these new services are implemented and run on cloud resources, i. e., general purpose hardware that are deployed as close to the customers as possible. We refer to the general purpose hardware close to the end user as *Microdatacenters* and we envision an additional service that can be offered by an ISP to significantly increase the flexibility and performance of general content delivery.

7.1 Microdatacenter Network Footprint

Since the new task is based on the concept of Microdatacenters, we first need to understand where they are best deployed. Most ISPs' networks consist of an access network to provide Internet access to DSL and/or cable customers, as well as an aggregation network for business and/or VPN customers. Routers at this level are often referred to as *edge routers*. The access and aggregation networks are then connected to the ISP's backbone which consists of *core routers*. *Border routers* are core routers that are used to connect either to other networks or to co-location centers. Opportunities to deploy Microdatacenters exist at each level: edge, core, or border router locations.

The advantage of deploying service infrastructure only at the core router locations is that there are a few large and well established locations. This is also a disadvantage as location diversity is limited and the servers can be quite far from the end-users. In contrast, location diversity is highest at the edge router locations. However, it might not be possible to deploy a Microdatacenter immediately at the locations, i.e., due to limited space and/or power at the facilities, or due to cost. These locations, however, minimize the distance to the customers and are therefore the most promising location for Microdatacenters. Border router locations are often a subset of core routers, hence they inherit the same advantages and disadvantages and are therefore not prime locations.

The advantage of using an ISP operated set Microdatacenters vs. a public cloud service is the chance to minimize the distance to the end-user. Moreover, if offers the possibility to directly control the location of the slices within the ISPs network while allocating and freeing them on demand. Thus, through cooperation with the ISP a good mapping of user demands to resource slices can be ensured.

7.2 Network Platform as a Service(NetPaaS)

Network Platform as a Service (NetPaaS) is a communication framework offered by an ISP and enables CDIs to use a hosting infrastructure based on Microdatacenters that scales or shrinks according to end-user demands. Figure 7.1 illustrates the concept on how this works with a CDI deploying an application. Ideally, the Microdatacenters are deployed close to the customers and can be found in every Point of Presence (PoP) of the ISP. A CDI wishing to deploy a an application can target specific Microdatacenters through NetPaaS and directly use the resources available at the location. Through this, the capital expenditures and operating costs are minimized, as well as the distance between the application and the source of the demand. For the ISP it means utilizing its already operational sites for further revenue, thus creating an incentive to actually deploy the Microdatacenters.

Any popular, Internet based application has no choice but to use an adaptive design, e.g., one that scales with the number of users. It is necessary to avoid being the victim of your own success, to be capable of handling flash-crowds [71], and to compensate for the limitations of current hardware. The motivation of third parties to deploy their applications to Microdatacenters managed through NetPaaS include (a) reducing infrastructure costs, (b) properly scaling and dimensioning the service

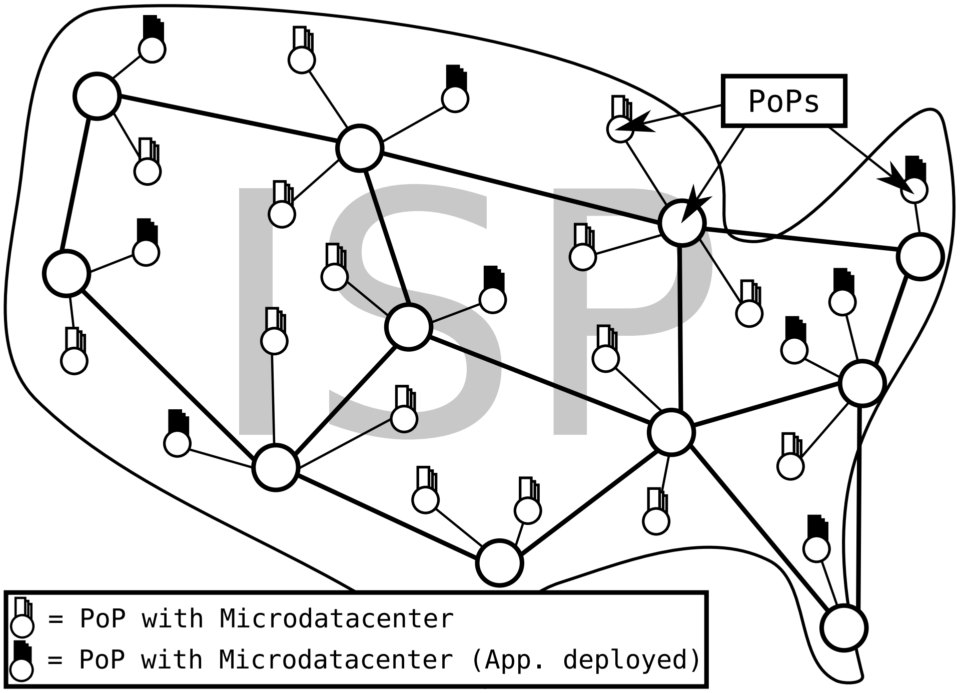

PoPs
ISP
= PoP with Microdatacenter
= PoP with Microdatacenter (App. deployed)

Figure 7.1: **Microdatacenters in an ISP with NetPaaS enabled**

(which is difficult to do before launching), and (c) improving the end-user experience, which can be a key factor that determines the success or failure of a service [77].

Sketching NetPaaS, we envision it to be a service offered by the ISP. It uses as its base unit of resource allocation the notion of a Microdatacenter *slice*. It is the ISP's task to allocate/de-allocate the slices since it operates the Microdatacenters. The third-party requests slices based on its clients demand and the resources needed to run its application. Once the slice is allocated, the application can be installed on the slice. From that point on, the CDI fully controls the operation of the application installed in the Microdatacenter. Negotiation about slices allocations are done via the NetPaaS Slice Allocation Interface through which third party client demands and application requirements are matched to the ISPs resources. However, to map demands to resources in an efficient manner is the task of the ISP and part can be easily reduced to the concept of PaDIS. In addition, we sketch the idea of an interface that allows for access to the billing information.

7.3 NetPaaS Building Blocks

We envision NetPaaS to comprise of three parts. First, the Slice Allocation is used to create, update, move and decommission slices of CDIs on the Microdatacenters.

Seconds, the User-Slice Assignment is based on PaDIS and CaTE to operate the slices assigned to CDIs efficiently in the network of the ISP. And third, the monitoring and billing offers CDIs direct control over the slices.

7.3.1 Slice Allocation

The Slice allocation enables a CDI and ISP to cooperate for discovery and allocating slices in Microdatacenters close to the end-user that are able to satisfy the demands. We envision a simple protocol that handles the dynamic deployment of CDI software and applications on Microdatacenters. It needs to be able to place the slices of the CDI on the Microdatacenters by using content demands as well as the network infrastructure. Furthermore, it needs to be fully automatic, as Microdatacenter deployment can be dynamic due to diurnal patterns, flash crowds or one time events. Note that this cooperation is only for deployment and management of the CDI deployment on the Microdatacenters, it is not used for the dynamic operation.

7.3.2 User-Slice Assignment

The purpose of the user-slice assignment interface is to enable small time scale interactions between the application and the ISP, to ensure that end-user demand is mapped to the appropriate slice. Therefore, the interface has to be integrated into the process used by the application to map user requests to slices.

Since slice allocated through NetPaaS can be seen as network location, we propose to use PaDIS to do the mapping of end-users to the slices (Section 5). Furthermore, by using PaDIS for the mapping, CaTE (Section 6) is automatically enabled in the ISPs network offering Microdatacenters through NetPaaS.

7.3.3 Monitoring and Billing

It is important for a CDI to track and minimize the cost of the use of Microdatacenters and NetPaaS. The first step towards this is that with every slice is annotated with a price. This can be transfered inline when using the Slice Allocation.

We expect that the billing of a slice allocated via NetPaaS follows that of large-scale datacenters. This means that there is an installation cost and a usage cost. The installation cost applies to a single slice in a Microdatacenter and is charged only once or over long time intervals, e. g., hours, and is fixed. The installation cost typically increases if additional licenses have to be leased, e. g., software licences. The installation cost can depend on the location of the Microdatacenter that hosts the slice or the time-of-day.

8 Related Work

Content delivery networks are used to provide fast and scalable commercial-grade Web applications [85]. CDIs rely on large-scale commodity infrastructures to improve application performance. Krishnamurthy et al. [78] and later Huang et al. [66] characterize their performance, i. e., by quantifying the end-user performance, analyzing the DNS redirection overhead, unveiling the CDI server location, and assessing their availability. Su et al. [127] propose to utilize CDIs' redirection to locate high performance paths. Choffnes et al. [23] propose to use information from CDIs to bias neighbor selection in P2P systems without any path monitoring or probing. Recently, Triukose et al. [129] show that most popular CDIs and several community CDIs serve any object from any server. This insight is used to show that the CDI's infrastructure can be used to amplify attacks against CDI costumer web sites. Our work leverages this observation by including ISP information for server selection and thus improves end-user performance and enables ISP traffic engineering.

One specific problem when introducing a new system is compatibility with the current state-of-the-art. In our case, there is rich literature for server selection [105]. Some of the proposals are distributed in nature [115, 135], others utilize the network view of the ISP [107]. A number of optimization techniques have been proposed by ISPS to improve user-server assignment by utilizing the service logs [102, 143] or by modifying the routing of user requests [79]. In many cases, the assignment of users to servers is done so as to minimize operational cost for bandwidth [3, 55] or energy [109]. Recently, game-theoretical approaches for ISP-CDI integration [70] and collaboration [37] for traffic engineering have been proposed. The IETF ALTO working group is examining solutions for the use case: mapping users to servers. Google is proposing a DNS extension [32] to improve end-user to server mapping.

The ideas for system design for PaDIS presented in this work build upon previous work on biasing peer selection in P2P systems [9, 10]. Our work also utilizes the insights from previous work [21] which has shown that server selection is important for enhancing the end-user experience. To the best of our knowledge, this was the first work proposing to deploy a system for direct collaboration of ISPs with third parties. We based PaDIS on the insight that today's content is usually accessible from multiple locations. In the past, game-theoretic studies [37, 70] have investigated the feasibility of ISP/CDI cooperation and the potential of an ISP deployed CDIs. However, no system to enable this was proposed.

With PaDIS, we tackled the question of how to meet the requirements of critical applications with stringent Service Level Agreements needed by ISPs. Today, ISPs rely on traffic engineering [17] to better control the IP packet flows. Several techniques have been proposed in the literature, some require tuning of routing protocols used inside the ISP network [47, 48, 133], while others rely on multipath [40, 42, 44, 55, 72, 140]. Changing routing weights can lead to oscillations [58] and is applied in time scales of hours. Multipath enables ISPs to dynamically distribute the traffic load in the presence of volatile and hard to predict traffic demand changes [40, 42, 44, 72], even at very small time scales, but requires additional configuration and management or router support.CaTE is complementary to both routing-based traffic engineering and multipath enabled networks.

Traffic engineering relies on the availability of information about traffic demands, which can be obtained either by direct observations [42, 43, 61, 131] or through inference [41, 93, 124, 141, 142]. CaTE relies on the network location diversity exposed by CDIs [7]. Game-theoretic results [37, 70, 89] show that the CDI/ISP collaboration can lead to a win-win situation. Recent studies also show that content location diversity has significant implications on ISP traffic engineering [120]. To our knowledge, CaTE is the first system that proposes to leverage the benefits of a direct CDI/ISP collaboration.

Finally, we turn to the emerging field of ISP operated cloud installations. In fact, Virtualization has received a lot of attention recently, as a flexible way to allocate heterogeneous resources, including computation, storage [136] (e. g., VMWare, Xen, and Linux VServer), network [116], in datacenters [16] and even in embarrassingly distributed clouds [25]. To capitalize on the flexibility and elasticity offered by virtualization, a number of systems have been built to automate data and server placement [4, 34, 134] and server migration [20, 83] even between geographically-distributed datacenters. The IETF CDNi working group is focusing on CDI interconnection to enable interoperability of multiple CDIs. Our proposed scheme of using Microdatacenters together with a unified protocol for allocation slices of these infrastructures through NetPaaS enable an even closer cooperation between ISPs and CDIs.

9 Conclusion

People value the Internet for the content and the applications it makes available [69]. For example, the demand for online entertainment and web browsing has exceeded 70% of the peak downstream traffic in the United States [114]. Recent traffic studies [54, 80, 107] show that a large fraction of Internet traffic is originated by a small number of Content Distribution Infrastructures (CDIs). Major CDIs are highly popular rich media sites such as YouTube and Netflix, One-Click Hosters (OCHs), e. g., RapidShare, as well as Content Delivery Networks (CDN) such as Akamai and hyper-giants, e. g., Google or Yahoo!. Gerber and Doverspike [54] report that a few CDIs account for more than half of the traffic of a US-based Tier-1 carrier.

To cope with the increasing demand for content, CDIs deploy massively distributed infrastructures [85] to replicate content and make it accessible from different locations in the Internet [7, 129]. But at the same time, they struggle to map users to the best server, regardless of whether the best server is the closest, the one with the most capacity or the lowest delay. The inability for CDIs to map end-users to the right server stems from the fact that CDIs have limited visibility into ISP networks as well as their setup, operation and current state.

One way to overcome the issue of mapping users to a close server is to use IP-Geolocation. By choosing a server that is geographically close, the performance for the end-user is good. But there is a downside; the assumption that the Geolocation information is correct in the first place. We tackle this indeterminate assumption in two ways. First, we analyze a connection log from a large, commercial CDN that supplies user generated geolocation information with each request. The analyzed service is mainly US centric and is therefore biased towards ISPs in North America.

In this analysis, we find that ISPs offering land line subscriptions show reasonable confidence that end-user location and IP-Geolocation are close to each other. In contrast, ISPs running mobile networks show no signs of the end-user location being close to the IP address being used. We find that, if IP-Geolocation is possible, it is usually possible to reliably determine the country or, in the case of the US, the state a user is in. However, narrowing the area down to a city is only possible in a few cases.

The second geolocation analysis broadens the scope from a deployment view by using already existing geolocation databases. We compare multiple databases, first to each other, and then to known ground truth from a large European ISP. We find that the geolocation databases themselves do not agree on the location of IP addresses. When comparing the databases to the ground truth information, we find that the mis-location of an IP address is in the order of country sizes. This observation is in line with the results from the previous end-user based analysis. Thus, we draw two conclusion: first, Geolocation databases generally manage to correctly identify a country, but fail at identifying a state or a city. And second, IP geolocation is not fit for aiding CDIs in finding close servers.

This observation shows that the core of the problem, i. e., that CDIs have limited visibility and insufficient knowledge about ISPs, has to be tackled a different way. In fact, ISPs have the knowledge that CDIs need, but lack a mechanism to supply it to CDIs. To this end, we introduce the Provider-aided Distance information System (PaDIS). The idea is simple: while selecting a server, a CDI finds all eligible servers that can handle the request. When the list is assembled, the question becomes one of the network state between the end-user requesting the content and the eligible servers. By enabling the CDI to query the network status through PaDIS, the ISP can use its intimate network knowledge to aid the CDI. Also, we show how PaDIS integrates into today's content delivery ecosystem while staying fully transparent to end-users.

Next, we turn to the evaluation of PaDIS. We use active measurement for different types of CDIs to simulate the effect of PaDIS on content delivery. First, we turn to a large, independent Content Distribution Network (CDN) and experiments based on active downloading of content couples with the operational network knowledge we obtained from a large European Tier-1 ISP. Our results show that, with PaDIS enabled, the server selection, in terms of end-user experience, can be significantly improved. In detail, we find that the download time, especially for small to medium sized objects, can be reduced by a factor of up to 3. Next, we turn our attention to the analysis of a popular One-Click Hoster (OCH). Here, we find that, with PaDIS, download times can be reduced by a factor of four during peak hours. Based on these results, we conclude that PaDIS is capable of optimizing CDI content delivery by reducing the time-to-download.

However, only improving content delivery for CDIs while not taking the ISPs into account hinders the acceptance of PaDIS. To this end, we propose and evaluate the concept of Content-aware Traffic Engineering, short CaTE, which leverages the fact that ISPs can, through PaDIS, influence the server selection of CDIs. In fact, by selecting a CDI server based on network criteria, PaDIS also selects the network path between the server and the end-user. This enables an ISP to perform highly flexible and dynamic traffic engineering on the CDI's traffic. Furthermore, CaTE is complimentary to the traditional link weights based traffic engineering. At the same time, CaTE does not interfere with protocol based flow control, as it merely changes the server from which the content is being fetched.

To evaluate the impact of CaTE on an ISP network, we rely on several datasets. First, we obtain the router level topology of a European Tier-1 ISP, its general traffic matrix as well an anonymized packet level trace spanning 10 days. We use this data together to simulate CaTE and its effects on the ISP's network. Our results show that CaTE is indeed able to significantly improve the traffic engineering and network operation of the ISP. In detail, we see that the overall traffic in the ISP's network drops by up to 6%, while the maximum link utilization is reduced by up to 60% at peak times. With regards to the CDI operations, we find the same behavior as before, specifically that the path length, as well as the delay between the server and the end-user is reduced. Furthermore, we show that the results of CaTE and PaDIS are not an artifact of the data we used, we repeat our study with two more networks, i. e., the AT&T network measured by Rocketfuel as well as the Abeline backbone. In these networks, we find that CaTE shows similar improvements. On one hand, CaTE is able to improve the network operation of the ISP, while on the other hand keeping benefits for CDIs and end-users due to its foundations in PaDIS.

Finally, we present the idea of combining PaDIS and CaTE with the dynamic deployment of CDI resources on ISP operated Microdatacenters, the CDI performance as well as CaTE can be pushed even further. Thus, we design another service, called Network Platform as a Service (NetPaaS), which allows CDIs and ISP to cooperate not only on the user assignment, but also to dynamically deploy and remove servers.

We believe that PaDIS and the concepts built upon it can play a significant role in the content delivery ecosystem. New CDIs and content providers can use PaDIS, CaTE and NetPaaS to enhance their current operation, establish targeted footprints at limited cost or deploy to completely new areas without effort. For ISPs, it is easy to see that the number of different resources as well as the number of CDIs that use them only increases their flexibility and agility in terms of network operations and traffic engineering. PaDIS, CaTE and NetPaaS are the tools for enabling Internet wide CDI-ISP collaboration for improved network operation and accelerated content delivery.

Acknowledgements

Over the course of my Ph.D. I met and worked with a lot of people. With some, I worked from the first to the last day, others I only collaborated with briefly. Some of you I have never met in person, but know only through email, Skype or phone. Every single one of you gave a different view or an insight that I missed and helped me complete this work. For this, I would like to thank every co-author, every collaborator, every co-worker and every proof reader for making this work possible.

Special thanks goes my advisor and mentor Anja Feldmann. It is a privilege to work with Anja, who sees the angles that we miss, the structure that we just could not get right, and has a always lent a helping hand when things were going tough.

I want to thank my very close collaborators, Steve Uhlig, George Smaragdakis and Benjamin Frank for always being creative with ideas and solutions. Without the discussion, fights, and different approaches that each of us brought to the team, this work would have never been possible.

I thank Bruce Maggs for sharing his vast knowledge of CDN operation and the ease with which he managed to transfer it. Not only do I owe a great deal of the CDN side of this work to him, I also want to thank him for helping me organize a very rewarding and successful internship.

I also want to thank Akamai Technologies, Inc. for having the privilege of interning there. Special thanks goes to the Custom Analytics group that I got to work with. KC, Steve, Jan, Rick, Arthur and David — thank you for a great time.

Last but not least, I want to thank everybody at FG-INET whom I had the pleasure of working with. I especially thank the secretaries, Britta and Birgit, and the infrastructure administrators for always being helpful, open and solving problems that I myself caused. Additionally, I want to thank the T-Labs, foremost Michael Düser, Vinay Aggrawal and Andreas Gladisch, for the collaboration with the FG-INET chair, without whom the ISP related work would have been much more difficult, if not impossible to achieve.

Finally, I want to thank everyone that helped me proof read and improve this work. Special thanks goes to my wife Jennifer, who managed to correct the entire thesis multiple times. Also, thanks to Oliver Holschke, Dan Levin, Oliver Hohlfeld and Bernhard Ager for giving valuable feedback.

List of Figures

List of Tables

Bibliography

[1] Acharya, H. B., and Gouda, M. G. The Theory of Network Tracing. In *Proceedings of ACM Symposium on Principles of Distributed Computing (PODC)* (2009).

[2] Aditya, P.and Zhao, M., Lin, Y., Haeberlen, A., Druschel, P., Maggs, B., and Wishon, B. Reliable Client Accounting for Hybrid Content-Distribution Networks. In *Proceedings of ACM Symposium on Networked Systems Design and Implementation (NSDI)* (2012).

[3] Adler, M., Sitaraman, R., and Venkataramani, H. Algorithms for Optimizing Bandwidth Costs on the Internet. In *IEEE Workshop on Hot Topics in Web Systems and Technologies* (2006).

[4] Agarwal, S., Dunagan, J., Jain, N., Saroiu, S., Wolman, A., and Bhogan, H. Volley: Automated Data Placement for Geo-Distributed Cloud Services. In *Proceedings of ACM Symposium on Networked Systems Design and Implementation (NSDI)* (2010).

[5] Ager, B., Chatzis, N., Feldmann, A., Sarrar, N., Uhlig, S., and Willinger, W. Anatomy of a Large European IXP. In *Proceedings of ACM SIGCOMM* (2012).

[6] Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S. Comparing DNS Resolvers in the Wild. In *Proceedings of ACM Internet Measurement Conference* (2010).

[7] Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S. Web Content Cartography. In *Proceedings of ACM Internet Measurement Conference* (2011).

[8] Ager, B., Schneider, F., Kim, J., and Feldmann, A. Revisiting Cacheability in Times of User Generated Content. In *IEEE Global Internet* (2010).

[9] Aggarwal, V., Akonjang, O., and Feldmann, A. Improving User and ISP Experience through ISP-aided P2P Locality. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2008).

[10] Aggarwal, V., Feldmann, A., and Scheideler, C. Can ISPs and P2P Users Cooperate for Improved Performance? *ACM Computer Communication Review 37*, 3 (2007).

[11] Akamai Inc. Akamai. `http://www.akamai.com`.

[12] Akamai, Inc. Akamai's EdgeScape. `http://www.akamai.com/html/technology/products/edgescape.html`.

[13] Akamai, Inc. Streamingmedia blog. http://blog.streamingmedia.com/the_business_of_online_vi/2011/06, 2011.

[14] Alimi, R., Penno, R., and Yang, Y. ALTO Protocol. draft-ietf-alto-protocol-12, 2011.

[15] Antoniades, D., Markatos, E. P., and Dovrolis, C. One-click Hosting Services: a File-sharing Hideout. In *Proceedings of ACM Internet Measurement Conference* (2009).

[16] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. Above the Clouds: A Berkeley View of Cloud Computing. UC Berkeley Technical Report EECS-2009-28, 2009.

[17] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and Xiao, X. Overview and Principles of Internet Traffic Engineering. RFC3272.

[18] Awerbuch, B., and Leighton, F. Multicommodity Flows: A Survey of Recent Research. In *International Symposium on Algorithms and Computation (ISAAC)* (1993).

[19] Azar, Y., Naor, J., and Rom, R. The Competitiveness of on-line Assignments. *Proceedings of ACM Symposium on Discrete Algorithms (SODA) 18*, 2 (1995).

[20] Bradford, R., Kotsovinos, E., Feldmann, A., and Schiöberg, H. Live Wide-Area Migration of Virtual Machines Including Local Persistent State. In *Proceedings of ACM Federated Computing Research Conference (FCRC)* (2007).

[21] Carter, R. L., and Crovella, M. E. On the Network Impact of Dynamic Server Selection. *Computer Networks 31*, (23-24) (1999).

[22] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y., and Moon, S. Analyzing the Video Popularity Characteristics of Large-scale User Generated Content Systems. *IEEE/ACM Transactions on Networks 17*, 5 (2009).

[23] Choffnes, D., and Bustamante, F. Taming the Torrent: a Practical Approach to Reducing Cross-ISP Traffic in Peer-to-peer Systems. In *Proceedings of ACM SIGCOMM* (2008).

[24] Chow, B., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., and Molina, J. Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control. In *ACM workshop on Cloud computing security* (2009).

[25] Church, K., Greenberg, A., and Hamilton, J. On Delivering Embarrasingly Distributed Cloud Services. In *ACM Workshop on Hot Topics in Networks (HotNets)* (2008).

[26] Cisco. Forecast and Methodology, 2011-2016. http://www.cisco.com.

[27] Cisco. NetFlow services and applications. White paper: http://www.cisco.com/warp/public/732/netflow, 1999.

[28] Cisco. Cariden MATE. https://marketplace.cisco.com/catalog/products/1581, 2013.

[29] Coffman, J. E., Garey, M., and Johnson, D. *Approximation Algorithms for Bin Packing: A Survey*. PWS Publishing Company, 1997.

[30] Cohen, B. Incentives Build Robustness in BitTorrent. In *P2PEcon Workshop* (2003).

[31] Coltun, R., Ferguson, D., and Moy, J. OSPF for IPv6. RFC2740, 1999.

[32] Contavalli, C., van der Gaast, W., Leach, S., and Rodden, D. Client IP Information in DNS Requests. IETF draft, work in progress, draft-vandergaast-edns-client-ip-01.txt, 2010.

[33] Cormode, G., and Hadjieleftheriou, M. Methods for Finding Frequent Items in Data Streams. *The International Journal on Very Large Data Bases (VLDB) 19*, 1 (2010).

[34] Cronin, E., Jamin, S., Jin, C., Kurc, A., Raz, D., and Shavitt, Y. Constraint Mirror Placement on the Internet. *IEEE Communication Magazine 20* (2002).

[35] Czumaj, A., Riley, C., and Scheideler, C. *Perfectly Balanced Allocation*. Springer, 2003.

[36] Dhungel, P., Ross, K. W., Steiner, M., Tian, Y., and Hei, X. Xunlei: Peer-Assisted Download Acceleration on a Massive Scale. In *Proceedings of ACM Passive and Active Measurement Conference (PAM)* (2012).

[37] DiPalantino, D., and Johari, R. Traffic Engineering versus Content Distribution: A Game-theoretic Perspective. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2009).

[38] Dobrian, F., Awan, A., Stoica, I., Sekar, V., Ganjam, A., Joseph, D., Zhan, J., and Zhang, H. Understanding the Impact of Video Quality on User Engagement. In *Proceedings of ACM SIGCOMM* (2011).

[39] Dreger, H., Feldmann, A., Mai, M., Paxson, V., and Sommer, R. Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In *Usenix Security Symposium* (2006).

[40] Elwalid, A., Jin, C., Low, S., and Widjaja, I. MATE : MPLS adaptive traffic engineering. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2001).

[41] Erramilli, V., Crovella, M., and Taft, N. An Independent-connection Model for Traffic Matrices. In *Proceedings of ACM Internet Measurement Conference* (2006).

[42] Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., and True, F. Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. *IEEE/ACM Transactions on Networks 9*, 3 (2001).

[43] Feldmann, A., Kammenhuber, N., Maennel, O., Maggs, B., De Prisco, R., and Sundaram, R. A Methodology for Estimating Interdomain Web Traffic Demand. In *Proceedings of ACM Internet Measurement Conference* (2004).

[44] Fischer, S., Kammenhuber, N., and Feldmann, A. REPLEX: Dynamic Traffic Engineering based on Wardrop Routing Policies. In *ACM Conference on emerging Networking Experiments and Technologies (CoNEXT)* (2006).

[45] Fischer, S., Räcke, H., and Vöcking, B. Fast Convergence to Wardrop Equilibria by Adaptive Sampling Methods. In *Proceedings ACM Symposium on Theory of Computing (STOC)* (2006).

[46] Fischer, S., and Vöcking, B. Adaptive Routing with Stale Information. In *Proceedings of ACM Symposium on Principles of Distributed Computing (PODC)* (2005).

[47] Fortz, B., and Thorup, M. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2000).

[48] Fortz, B., and Thorup, M. Optimizing OSPF/IS-IS Weights in a Changing World. *IEEE Journal on Selected Areas in Communications 20* (2002).

[49] Francois, P., and Bonaventure, O. Avoiding Transient Loops during the Convergence of Link-State Routing Protocols. *IEEE/ACM Transactions on Networks 15* (2007).

[50] Freedman, M. J. Experiences with CoralCDN: A Five-Year Operational View. In *Proceedings of ACM Symposium on Networked Systems Design and Implementation (NSDI)* (2010).

[51] Freedman, M. J., Vutukurum, M., Feamster, N., and Balakrishnan, H. Geographic Locality of IP Prefixes. In *Proceedings of ACM Internet Measurement Conference* (2005).

[52] GeoBytes Inc. GeoNetMap - geobytes' IP address to geographic location database. `http://www.geobytes.com/GeoNetMap.htm`.

[53] GeoURL. The GeoURL ICBM Address Server. `http://www.geourl.org`.

[54] Gerber, A., and Doverspike, R. Traffic Types and Growth in Backbone Networks. In *Proceedings of IEEE Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)* (2011).

[55] Goldenberg, D., Qiuy, L., Xie, H., Yang, Y., and Zhang, Y. Optimizing Cost and Performance for Multihoming. In *Proceedings of ACM SIGCOMM* (2004).

[56] Google, Inc. GoogleCache. `http://ggcadmin.google.com/ggc`.

[57] Graham, R. Bounds on Multiprocessing Timing Anomalies. *SIAM J. Applied Math.* (1969).

[58] Griffin, T., and Wilfong, G. On the Correctness of iBGP Configuration. In *Proceedings of ACM SIGCOMM* (2002).

[59] Gueye, B., Uhlig, S., and Fdida, S. Investigating the Imprecision of IP Block-Based Geolocation. In *Proceedings of ACM Passive and Active Measurement Conference (PAM)* (2007).

[60] Gueye, B., Ziviani, A., Crovella, M., and Fdida, S. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networks 14*, 6 (2006).

[61] Gunnar, A., Johansson, M., and Telkamp, T. Traffic Matrix Estimation on a Large IP Backbone: A Comparison on Real Data. In *Proceedings of ACM Internet Measurement Conference* (2004).

[62] Hexasoft Development Sdn. Bhd. IP Address Geolocation to Identify Website Visitor's Geographical Location. http://www.ip2location.com.

[63] Host IP. My IP Address Lookup and GeoTargeting Community Geotarget IP Project. http://www.hostip.info.

[64] Hu, Z., and Heidemann, J. Towards Geolocation of Millions of IP Addresses. In *Proceedings of ACM Internet Measurement Conference* (2012).

[65] Huang, C., Li, J., Wang, A., and Ross, K. W. Understanding Hybrid CDN-P2P: Why Limelight Needs its Own Red Swoosh. In *Proceedings of ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV)* (2008).

[66] Huang, C., Wang, A., Li, J., and Ross, K. Measuring and Evaluating Large-scale CDNs. In *Proceedings of ACM Internet Measurement Conference* (2008).

[67] Inc., A. SureRoute. www.akamai.com/dl/feature_sheets/fs_edgesuite_sureroute.pdf.

[68] IP InfoDB. Free IP Address Geolocation Tools. http://ipinfodb.com/.

[69] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., and Braynard, R. Networking Named Content. In *ACM Conference on emerging Networking Experiments and Technologies (CoNEXT)* (2009).

[70] Jiang, W., Zhang, R.-S., Rexford, J., and Chiang, M. Cooperative Content Distribution and Traffic Engineering in an ISP Network. In *Proceedings of ACM SIGMETRICS* (2009).

[71] Jung, J., Krishnamurthy, B., and Rabinovich, M. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. In *World Wide Web Conference* (2002).

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

[73] Katz-Bassett, E., John, J., Krishnamurthy, A., Wetherall, D., Anderson, T., and Chawathe, Y. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of ACM Internet Measurement Conference* (2006).

[74] Key, P., Massoulié, L., and Towsley, D. Path Selection and Multipath Congestion Control. *Communication of the ACM 51* (2011).

[75] Khachiyan, L. A Polynomial Time Algorithm for Linear Programming. *Doklady Akademii Nauk SSSR* (1979).

[76] Knuth, D. E. *The Art of Computer Programming, Volume 2 (3rd Edition): Seminumerical Algorithms*. Addison-Wesley, 1997.

[77] Kohavi, R., Henne, R. M., and Sommerfield, D. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. In *KDD* (2007).

[78] Krishnamurthy, B., Wills, C., and Zhang, Y. On the Use and Performance of Content Distribution Networks. In *Proceedings of ACM Internet Measurement Conference* (2001).

[79] Krishnan, R., Madhyastha, H., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., and Gao, J. Moving Beyond End-to-end Path Information to Optimize CDN Performance. In *Proceedings of ACM Internet Measurement Conference* (2009).

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

[81] Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E., and Taft, N. Structural Analysis of Network Traffic Flows. In *Proceedings of ACM SIGMETRICS* (2004).

[82] Laoutaris, N., Sirivianos, M., Yang, X., and Rodriguez, P. Inter-Datacenter Bulk transfers with NetStitcher. In *Proceedings of ACM SIGCOMM* (2011).

[83] Laoutaris, N., Smaragdakis, G., Oikonomou, K., Stavrakakis, I., and Bestavros, A. Distributed Placement of Service Facilities in Large-Scale Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2007).

[84] Laoutaris, N., Smaragdakis, G., Rodriguez, P., and Sundaram, R. Delay Tolerant Bulk Data Transfers on the Internet. In *Proceedings of ACM SIGMETRICS* (2009).

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

[86] Lenstra, J., Shmoys, D., and Tardos, E. Approximation Algorithms for Scheduling Unrelated Parallel Machines. *ACM Journal on Mathematical Programming 46* (1990).

[87] Li, A., Yang, X., Kandula, S., and Zhang, M. CloudCmp: Comparing Public Cloud Providers. In *Proceedings of ACM Internet Measurement Conference* (2010).

[88] Liu, H. H., Wang, Y., Yang, Y., Wang, H., and Tian, C. Optimizing Cost and Performance for Content Multihoming. In *Proceedings of ACM SIGCOMM* (2012).

[89] Ma, R. T. B., Chiu, D. M., Lui, J. C. S., Misra, V., and Rubenstein, D. On Cooperative Settlement Between Content, Transit, and Eyeball Internet Service Providers. *IEEE/ACM Transactions on Networks 19* (2011).

[90] Maier, G., Feldmann, A., Paxson, V., and Allman, M. On Dominant Characteristics of Residential Broadband Internet Traffic. In *Proceedings of ACM Internet Measurement Conference* (2009).

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

[92] MaxMind LLC. `http://www.maxmind.com`.

[93] Medina, A., Taft, N., Salamatian, K., Bhattacharyya, S., and Diot, C. Traffic Matrix Estimation: Existing Techniques and New Directions. In *Proceedings of ACM SIGCOMM* (2002).

[94] Miniwats Marketing Group. Internet world stat: Usage and population statistics, 2009. `http://www.internetworldstats.com`.

[95] Mockapetris, P. Domain names - implementation and specification. RFC1035, 1987.

[96] Moy, J. OSPF Version 2. RFC2328, 1998.

[97] Navstar. GPS Standard Positioning Service (SPS) Performance Standard, version 4. `http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf`, 2008.

[98] Net World Map. The Net World Map Project. `http://www.networldmap.com`.

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

[100] Niven-Jenkins, B., Le Faucheur, F., and Bitar, N. Content Distribution Network Interconnection (CDNI) Problem Statement. RFC 6707, 2012.

[101] Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and Previdi, S. Use Cases for ALTO within CDNs. draft-jenkins-alto-cdn-use-cases-03, 2012.

[102] Nygren, E., Sitaraman, R. K., and Sun, J. The Akamai Network: A Platform for High-performance Internet Applications. *ACM SIGOPS Operating System Review 44* (2010).

[103] Oran, D. OSI IS-IS Intra-domain Routing Protocol. RFC1142, 1990.

[104] Padmanabhan, V. N., and Subramanian, L. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of ACM SIGCOMM* (2001).

[105] Pan, J., Hou, Y. T., and Li, B. An Overview of DNS-based Server Selections in Content Distribution Networks. *Computer Networks 43*, 6 (2003).

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

[107] Poese, I., Frank, B., Ager, B., Smaragdakis, G., and Feldmann, A. Improving Content Delivery using Provider-Aided Distance Information. In *Proceedings of ACM Internet Measurement Conference* (2010).

[108] Quova Inc. GeoPoint - IP geolocation experts. http://www.quova.com.

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

[110] Rekhter, Y., Li, T., and Hares, S. A Border Gateway Protocol 4 (BGP-4). RFC4271, 2006.

[111] Rosen, E., Viswanathan, A., and Callon, R. Multiprotocol Label Switching Architecture. RFC3031, 2001.

[112] Roughan, M. Simplifying the Synthesis of Internet Traffic Matrices. *ACM Computer Communication Review 35*, 5 (2005).

[113] Sandvine Inc. 2009 Global Broadband Phenomena. http://www.sandvine.com/news/global_broadband_trends.asp.

[114] Sandvine Inc. Global Broadband Phenomena. Research Report http://www.sandvine.com/news/global_broadband_trends.asp, 2011.

[115] Scellato, S., Mascolo, C., Musolesi, M., and Crowcroft, J. Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades. In *World Wide Web Conference* (2011).

[116] Schaffrath, G., Werle, C., Papadimitriou, P., Feldmann, A., Bless, R., Greenhalgh, A., Wundsam, A., Kind, M., Maennel, O., and Mathy, L. Network Virtualization Architecture: Proposal and Initial Prototype. In *ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA)* (2009).

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

[118] Schulze, H., and Mochalski, K. Internet Study 2008-9. http://www.ipoque.com/resources/internet-studies, 2009.

[119] Seedorf, J., and Burger, E. W. Application-Layer Traffic Optimization (ALTO) Problem Statement. RFC 5693, 2009.

[120] Sharma, A., Mishra, A., Kumar, V., and Venkataramani, A. Beyond MLU: An application-centric comparison of traffic engineering schemes. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2011).

[121] Sherwood, R., Bender, A., and Spring, N. DisCarte: A Disjunctive Internet Cartographer. In *Proceedings of ACM SIGCOMM* (2008).

[122] Siwpersad, S., Gueye, B., and Uhlig, S. Assessing the Geographic Resolution of Exhaustive Tabulation for Geolocating Internet Hosts. In *Proceedings of ACM Passive and Active Measurement Conference (PAM)* (2008).

[123] Software 77. Free IP to Country Database. http://software77.net/geo-ip/.

[124] Soule, A., Nucci, A., Cruz, R., Leonardi, E., and Taft, N. How to Identify and Estimate the Largest Traffic Matrix Elements in a Dynamic Environment. In *Proceedings of ACM SIGMETRICS* (2004).

[125] Spring, N., Mahajan, R., Wetherall, D., and Anderson, T. Measuring ISP Topologies with Rocketfuel. *IEEE/ACM Transactions on Networks 12*, 1 (2004).

[126] Spring, N., Wetherall, D., and Anderson, T. Scriptroute: A Public Internet Measurement Facility. In *Proceedings of USENIX USITS Symposium* (2003).

[127] Su, A., Choffnes, D. R., Kuzmanovic, A., and Bustamante, F. E. Drafting behind Akamai: Inferring Network Conditions based on CDN Redirections. *IEEE/ACM Transactions on Networks 17*, 6 (2009).

[128] Tippenhauer, N. O., Rasmussen, K. B., Pöpper, C., and Capkun, S. iPhone and iPod Location Spoofing: Attacks on Public WLAN-based Positioning Systems. Tech. rep., SysSec Technical Report. ETH Zurich, 2008.

[129] Triukose, S., Al-Qudah, Z., and Rabinovich, M. Content Delivery Networks: Protection or Threat? In *European Symposium on Research in Computer Security (ESORICS)* (2009).

[130] Triukose, S., Ardon, S., Mahanti, A., and Seth, A. Geolocating IP addresses in cellular data networks. In *Proceedings of ACM Passive and Active Measurement Conference (PAM)* (2012).

[131] Uhlig, S., Quoitin, B., Lepropre, J., and Balon, S. Providing Public Intradomain Traffic Matrices to the Research Community. *ACM Computer Communication Review 36*, 1 (2006).

[132] Valancius, V., Lumezanu, C., Feamster, N., Johari, R., and Vazirani, V. V. How Many Tiers? Pricing in the Internet Transit Market. In *Proceedings of ACM SIGCOMM* (2011).

[133] Wang, Y., Wang, Z., and Zhang, L. Internet Traffic Engineering Without Full Mesh Overlaying. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2001).

[134] Wang, Y. A., Huang, C., Li, J., and Ross, K. W. Estimating the Performance of Hypothetical Cloud Service Deployments: A Measurement-based Approach. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2011).

[135] Wendell, P., Jiang, J. W., Freedman, M. J., and Rexford, J. DONAR: Decentralized Server Selection for Cloud Services. In *Proceedings of ACM SIGCOMM* (2010).

[136] Whiteaker, J., Schneider, F., and Teixeira, R. Explaining Packet Delays under Virtualization. *ACM Computer Communication Review 41*, 1 (2011).

[137] Wong, B., Stoyanov, I., and Sirer, E. G. Geolocalization on the Internet Through Constraint Satisfaction. In *Proceedings of USENIX WORLDS Workshop* (2005).

[138] Xie, H., Yang, Y. R., Krishnamurthy, A., Liu, Y. G., and Silberschatz, A. P4P: Provider Portal for applications. In *Proceedings of ACM SIGCOMM* (2008).

[139] Yang, X., and de Veciana, G. Service Capacity of Peer to Peer Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)* (2004).

[140] Zhang, M., Yi, C., Liu, B., and Zhang, B. GreenTE: Power-aware Traffic Engineering. In *International Conference on Network Protocols* (2010).

[141] Zhang, Y., Roughan, M., Duffield, N., and Greenberg, A. Fast Accurate Computation of Large-scale IP Traffic Matrices from Link Loads. In *Proceedings of ACM SIGMETRICS* (2003).

[142] Zhang, Y., Roughan, M., Lund, C., and Donoho, D. An Information-theoretic Approach to Traffic Matrix Estimation. In *Proceedings of ACM SIGCOMM* (2003).

[143] Zhang, Z., Zhang, M., Greenberg, A., Hu, Y. C., Mahajan, R., and Christian, B. Optimizing Cost and Performance in Online Service Provider Networks. In *Proceedings of ACM Symposium on Networked Systems Design and Implementation (NSDI)* (2010).