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

# Dynamic Content Delivery Infrastructure Deployment using Network Cloud Resources

vorgelegt von
Benjamin Frank (M.Sc.)
aus Oldenburg

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 M. Maggs, Ph. D., Duke University, NC, USA |
| Gutachter: | Prof. Steve Uhlig, Ph. D., Queen Mary, University of London, UK |
| Gutachter: | Georgios Smaragdakis, Ph. D., Technische Universität Berlin, Germany |

Tag der wissenschaftlichen Aussprache: 16. Dezember 2013

Berlin 2014
D 83

# Eidesstattliche Erklärung

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

Datum Benjamin Frank (M.Sc.)

# Abstract

Millions of people value the Internet for the content and the applications it makes available. To cope with the increasing end-user demand for popular and often high volume content, e.g., high-definition video or online social networks, massively distributed *Content Delivery Infrastructures (CDIs)* have been deployed.

However, a highly competitive market requires CDIs to constantly investigate new ways to reduce operational costs and improve delivery performance. Today, CDIs mainly suffer from limited agility in server deployment and are largely unaware of network conditions and precise end-user locations, information that improves the efficiency and performance of content delivery. While newly emerging architectures try to address these challenges, none so far considered *collaboration*, although ISPs have the information readily at hand.

In this thesis, we assess the impact of collaboration on content delivery. We first evaluate the design and operating space of todays content delivery landscape and quantify possible benefits of collaboration by analyzing operational traces from an European Tier-1 ISP. We find that collaboration when assigning end-users to servers highly localizes CDI traffic and improves end-user performance. Moreover, we find significant path diversity which enables new mechanisms for traffic management.

We propose two key enablers, namely *in-network server allocation* and *informed user-server assignment*, to facilitate CDI-ISP collaboration and present our system design, called NetPaaS (Network Platform as a Service), that realizes them. In-network server allocation offers agile server allocation close to the ISPs end-users leveraging virtualization technology and cloud style resources in the network. Informed user-server assignment enables ISPs to take network bottlenecks and precise end-user locations into account and to recommend the best possible candidate server for individual end-users to CDIs. Therefore, NetPaaS provides an additional degree of freedom to scale-up or shrink the CDI footprint on demand.

To quantify the potential of collaboration with NetPaaS, we perform a first-of-its-kind evaluation based on operational traces from the largest commercial CDI and an European Tier-1 ISP. Our findings reveal that dynamic server allocation based on accurate end-user locations and network conditions enables the CDI to better cope with increasing and highly volatile demand for content and improves the end-users performance. Moreover, recommendations from NetPaaS result in better utilization of existing server infrastructure and enables the ISP to better manage traffic flows inside its network.

We conclude, that NetPaaS improves the performance and efficiency of content delivery architectures while potentially reducing the required capital investment and operational costs. Moreover, NetPaaS enables the ISP to achieve traffic engineering goals and therefore offers a true win-win situation to both CDIs and ISPs.

# Zusammenfassung

Millionen von Menschen schätzen die Inhalte und Anwendungen, die das Internet zur Verfügung stellt. Um der steigenden Nachfrage an populären Inhalten wie z.B. High-Definition Video oder Online Social Networks nachzukommen, wurden weit verteilte *Content Delivery Infrastructures (CDIs)* aufgebaut.

Damit CDIs im harten Wettbewerbs bestehen können, suchen sie ständig neue Möglichkeiten um laufende Kosten zu senken und Ihre Leistungsfähigkeit zu steigern. Jedoch machen den CDIs eine geringe Agilität bei der Allokation von Servern zu schaffen. Informationen zur Steigerung von Effizienz und Leistungsfähigkeit wie z.B. aktuelle Netzwerkbedingungen und präzise User-Positionen sind den CDIs unbekannt. Obwohl Internet Service Provider (ISPs) diese Informationen besitzen, lassen auch neuere CDI-Architekturen eine mögliche *Kollaboration* außer Acht.

Diese Dissertation untersucht den Einfluss von Kollaboration auf Content Delivery. Zunächst wird das heutige Design- und Betriebsfeld untersucht. Eine Analyse der operativen Daten eines Europäischen Tier-1 ISPs erörtert mögliche Verbesserungen. Erste Ergebnisse zeigen, dass Kollaboration bei der Zuordnung von Usern zu CDI Servern den Netzwerkverkehr lokal begrenzt und die Geschwindigkeit erhöht. Vorhandene Netzwerkpfade eröffnen neue Möglichkeiten der Verkehrssteuerung.

Um die Kollaboration zwischen CDIs und ISPs zu ermöglichen, beschreibt diese Arbeit die beiden Key Enabler *In-Network Server Allocation* und *Informed User-Server Assignment*. Sie stellt außerdem ein Systemdesign vor, das diese realisiert: *NetPaaS* (Network Platform as a Service). In-Network Server Allocation nutzt im ISP verteilte Resourcen und aktuelle Virtualisierungstechnologien um eine agile Serverallokation zu ermöglichen. Informed User-Server Assignment erlaubt es ISPs, mögliche Netzwerkengpässe und präzise User-Positionen einzukalkulieren und so CDIs den besten Server für individuelle Nutzer zu empfehlen. Damit bietet NetPaaS einen zusätzlichen Freiheitsgrad zur dynamischen Skalierung von Serverinfrastrukturen.

Um das Kollaborationspotential von NetPaaS aufzuzeigen, wird erstmals eine Studie mit operativen Daten des größten kommerziellen CDI und einem Europäischen Tier-1 ISP durchgeführt. Die Ergebniss zeigen, dass eine auf präzisen User-Positionen und aktuellen Netzwerkbedingungen basierende dynamische Serverallokation es dem CDI ermöglicht, besser mit der stark schwankenden Nachfrage nach Inhalten zurecht zu kommen und die Geschwindigkeit der Nutzer zu verbessern. Darüber hinaus führt die Nutzung von NetPaaS zu einer besseren Auslastung vorhandener Serverinfrastrukturen und ermöglicht ein verbessertes Verkehrsmanagement im Netz des ISP.

Diese Ergebnisse lassen den Schluss zu, dass NetPaaS die Leistungsfähigkeit und Effizienz von CDIs stark verbessert und unter Umständen laufende Kosten und Investitionen reduziert. NetPaaS verbessert weiterhin das Verkehrsmanagement des ISP und bietet somit eine echte “win-win” Situation für CDIs und ISPs.

# Publications

## Pre-published Papers

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

### Book Chapters

Frank, B., Poese, I., Smaragdakis, G., Feldmann, A., Maggs, B. M., Uhlig, S., Aggarwal, V., and Schneider, F. Collaboration Opportunities for Content Delivery and Network Infrastructures. *ACM SIGCOMM ebook on Recent Advances in Networking* (2013)

### International Conferences

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

Frank, B., Poese, I., Smaragdakis, G., Uhlig, S., and Feldmann, A. Content-aware Traffic Engineering. In *ACM SIGMETRICS* (2012)

### Peer-reviewed Journals

Poese, I., Frank, B., Ager, B., Smaragdakis, G., Uhlig, S., and Feldmann, A. Improving Content Delivery with PaDIS. *IEEE Internet Computing 16*, 3 (2012)

Poese, I., Frank, B., Smaragdakis, G., Uhlig, S., Feldmann, A., and Maggs, B. M. Enabling Content-aware Traffic Engineering. *ACM SIGCOMM Computer Communication Review 42*, 5 (2012)

Frank, B., Poese, I., Lin, Y., Smaragdakis, G., Feldmann, A., Maggs, B. M., Rake, J., Uhlig, S., and Weber, R. Pushing CDN-ISP Collaboration to the Limit. *ACM SIGCOMM Computer Communication Review 43*, 3 (2013)

## Workshops and Poster Sessions

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

## Technical Reports

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

# Contents

# 1
# Introduction

"Content is King" [33]: Predicted by Bill Gates in an essay from 1996, this quote has become the latest buzz in the Internet economy [7, 43, 49, 98, 137, 151]. User demand for popular and often high volume applications such as high-definition video, music, cloud-gaming, online social networks, and online-gaming is phenomenal; unbroken since years [71, 98, 151] and still expected to grow [43]. For example, the demand for online entertainment and web browsing is contributing 70% of the peak downstream traffic in the United States [151].

Recent studies [71, 98, 137] find that today's Internet traffic is dominated by content delivered by a variety of *Content Delivery Infrastructures* (CDIs). Major CDIs include highly popular Video Service Providers (VSPs), such as YouTube [38], Netflix [2], One-Click Hosters (OCHs), such as RapidShare [21] or Dropbox [60], as well as Content Delivery Networks (CDNs), such as Akamai [57,128], Limelight [106], and other hyper-giants, such as Google [26], Yahoo! or Microsoft [117]. Other popular and traffic heavy services using CDIs include music downloads and streaming (e.g., Pandora, iTunes, Spotify), cloud gaming (e.g., OnLive, PlayStation Netwok, Xbox One), Online Social Networks (OSNs, e.g., Facebook, Twitter or Google+), as well as online gaming (e.g., World of Warcraft, Farmville, Xbox Live).

Gerber and Doverspike [71] report that a hand full of *Content Delivery Infrastructures* (CDIs) are responsible for more than half of the traffic in North America. Poese et al. report similar observations for traffic of an European Tier-1 carrier. Labovitz [49] reports that 50% of North American traffic originates from just 35 sites/services with only a handful CDIs serving the traffic. In a previous study Labovitz et al. [98] infer that more than 10% of the total Internet inter-domain traffic originates from

Google, and Akamai claims to deliver more than 20% of the total Web traffic in the Internet [128]. Netflix, a company offering high definition video on-demand streaming, is responsible for a significant fraction of the traffic in North American ISPs during peak hour [151].

## 1.1 Challenges in Content Delivery

Even decades after the first commercial Content Delivery Infrastructures have been launched, the challenges content delivery still faces today are manifold. The question where to deploy additional server resources – and how much – is by no means easy to answer [66, 95]. The end-user demand for content is highly volatile, both spatial and temporal, and precisely locating end-users network positions turns out to be a tedious and error prone task [137, 141]. Novel and agile deployment strategies are required to further improve the CDIs performance and capacity as current approaches take up to multiple months and requires high capital investment. In the following, we discuss the challenges content delivery faces today in more detail.

### Infrastructure Deployment

To cope with the continuously growing end-user demand for content, CDIs use and continue to deploy massively distributed server infrastructures that replicate and distribute popular content in many different locations on the Internet [7, 102]. This implies that the deployment of server infrastructure is a challenge for CDIs. However, different players in the content delivery business have developed different strategies to handle the challenges in server deployment. As described by Tom Leigthon [102], these approaches include (1) centralized hosting, (2) datacenter based CDIs, (3) highly distributed cache-based CDIs, and (4) Peer-to-Peer (P2P) networks.

The first approach may be sufficient for small services targeted at local audiences and can be extended by geographical disperse mirrors. This improves the end-users performance as a server is closer to some of the users, improves scalability due to more servers being able to serve more end-users, and enhances reliability through redundancy. But the complexity of managing capacities, content replication as well as the financial investment for infrastructure deployment, hard to predict and highly volatile traffic levels in combination with the inability to absorb sudden demand surges, often referred to as *flashcrowds*, have paved the way for approaches 2 and 3. Both offer increased scalability and reliability by offloading the delivery of content from the original server onto a larger network of caches which are shared by numerous services and operated by a third party, the CDI.

Data center based CDIs leverage economy of scale over centralized hosting by operating a number of big data centers with thousands of server connected to hundreds of networks. While offering improved performance the gains are limited as the distance to end-users is still large according to any metric: the biggest 30 networks combined host roughly 50% of the end-users and the numbers decline very fast resulting in a long tail distribution of end-user over all the Internets networks to where a dedicated connection is economically infeasible for the CDI. As a result the traffic needs to cross many "middle mile" networks to reach a significant number of end-users even if the CDI connects to all large Tier-1 backbone networks. Another drawback of this architecture is the large network load that these datacenters impose on transit networks.

Other CDIs try to avoid these issues by deploying highly distributed cache servers in many different networks, mainly big eyeball ISPs (that host many end-users) and highly connected Tier-1 networks (which can also act as backups for smaller networks that do not host a CDI cache). While this deployment strategy solves the server to end-user distance problem the deployment itself is more complex and thus most likely more costly and time consuming. Because each network becomes a contractual partner, the CDI has, depending on the geographical location the network operates in, to take for example state regulations (e.g., telecommunication acts) or national standard bodies (e.g., for power standards) into account.

Last but not least, P2P networks rely on a huge number of end users to store, replicate, and distribute content. As a result P2P networks capacity scales with each user participating. It has been shown to scale well even in case of extreme flash crowds [161].

To name a few examples: one of the largest players in the content delivery business, Akamai, utilizes a highly distributed server infrastructure and operates more than 127,000 servers in 81 countries distributed across more than 1,150 networks [12, 128]. Google reportedly operates tens of data centers and front-end server clusters worldwide [76,96,168]. Microsoft has deployed its content delivery infrastructure in 24 locations around the world [117]. Amazon maintains at least 5 large data centers and caches in at least 21 locations around the world [19]. Limelight also utilizes a data center based deployment and operates thousands of servers in more than 22 delivery centers and connects directly to 600 networks worldwide [106].

### End-User to Server Assignment

A key component of any CDI is the assignment of end-users to servers (or peers in the case of P2P). The ability to assign end-users to servers on small timescale, e.g., in the order of minutes or even tens of seconds, is crucial for CDIs to react to sudden demand surges (flash crowds) and demands shifting from one network to another (regional shift). The assignment strategy of CDIs is also highly relevant with regards to economical aspects of content delivery. The CDI has to resolve the following trade-off: which server delivers the best performance for the end-user while offering the highest economical return for the CDI[1]. This decision includes various important parameters, such as server load, precise network location of end-users, and network conditions (e.g., network bottlenecks or peering cost), some of which require extensive but error prone measurements [3,128,166] by either the CDI or the end-users.

Today three main mechanism are used for the assignment of end-users to servers: (1) DNS based redirection, (2) HTTP redirection and (3) IP Anycast. The first solution leverages the fact that before an end-user establishes a connection it resolves a hostname using the Domain Name System (DNS). By transferring the administrative authority of a domain, or more often a subdomain, the CDI is responsible to resolve the hostname. It then is in the position to choose which of the available servers should answer the end-users request. The second solution uses redirection directives included in the HTTP protocol [32]. The main benefit of this solution is the additional information contained in the HTTP request, e.g., the requested object and the end-users IP address, but incurs at least one additional round trip time (RTT) and TCP handshake when a redirection is necessary, as the end-user has to establish a new TCP connection to the new server. The third solution delegates the issue of server selection to the routing layer of the end-users network. This solution has nearly no control over the server selection anymore. We will discuss the details of the drawbacks and benefits of end-user to server assignment methods in Chapter 2.4.3.

### Content Delivery Alliances

Although some CDI deployments already have a large global footprint, even the biggest players are still improving it and need deployment strategies for content delivery. Recently Akamai formed content delivery strategic alliances with major ISPs, including AT&T [11], Orange [14], Swisscom [15], and KT [13] to reduce network-related costs and improve network efficiency by outsourcing the hardware deployment and maintenance to said network operators. Google offers eyeball networks, that experience high peak traffic from Google's network, the opportunity to host one or more Google Global Caches (GGCs) [26, 75]. Those application specific caches will serve popular Google content including the traffic heavy YouTube video service. Thus, they offer traffic reductions and reduced network utilization to the network operator and improve the performance of the end-users. Netflix, while heavily relying on multiple CDIs, including Limelight and Level3, to deliver its high traffic volumes [2], recently announced to deploy its own content delivery infrastructure, called Open Connect [122], offering network operators that host the free of charge appliance potentially huge traffic reductions while improving the end-user quality of experience. Interestingly enough, Labovitz [49] found that while those servers are located in many networks none of them is a Tier-1 provider.

[1] In the case of P2P the economical return encompasses anything that increases the systems total capacity, e.g., faster download times and higher throughput

The combined efforts of CDIs and network operators clearly marks a paradigm shift in how content delivery infrastructure is deployed and opens up new possibilities for innovative approaches that foster collaboration between CDIs and network operators to take advantage of the business opportunities. After decades of ever increasing deployment for scalability, performance and cost issues, CDIs start noticing the limits in expanding their network footprint. Hereby, we stress that these are often not technical limits, but more business constraints and/or management overhead. In this context the formation of alliances seems to be the natural evolution of the content delivery business.

### Deployment Agility

Unfortunately, the deployment of servers that can satisfy the growing demand while providing good performance to end-users is a complex and tedious task. Finding the right locations to place additional servers without knowledge about the network and its traffic dynamics takes a significant amount of time and is prone to errors and inaccuracies. The necessary business arrangements also require time and effort, as every party wants to get the best possible deal to reduce cost and/or increase revenues. But even when the bargaining is done more time is required to commission the hardware, ship it to the agreed location, physically hook up the servers and connect it to the network. Depending on eventual Service Level Agreements (SLAs) the network operator might need additional time to configure the necessary network devices, e.g., routers, firewall, or intrusion detection systems. Last but not least the CDI's operations team has to install and configure the required software and once the server is fully functional and ready for operation the assignment strategy can include the newly deployed machine.

While some of the steps can be done in parallel to speed up the process, the initial search for a suited location and the resulting negotiations take most of the time that can span multiple months limiting the CDIs agility in server deployment [128]. Yet, the deployment is not the only aspect where more flexibility is needed. Once a server is deployed, the physical location of the hardware stays and the negotiated contracts are in place for a longer periods of time, e.g., tens of months to multiple years. This is because for the network operator frequent changes to the network configuration, e.g., physically removing and shipping hardware, updating security policies or possible routing changes, are highly inconvenient and disrupt normal operations. Also the involved re-negotiations impose a high burden on the involved business units of both the network operator and the CDI.

Most Content Delivery Infrastructures can handle additions and removal of servers easily, yet shipping around the hardware and reconfiguration of the software means that the additional resources are not available during that time resulting in paid but unusable capacities. Thus, altogether the situation for both, the CDIs and the network operators, are mediocre at best. While the movement of physical hardware and the resulting network changes should be kept to a minimum to ensure proper network operations it also limits the CDIs ability to react to increasing traffic demands and changes in traffic demand patterns in a timely fashion. This in turn increases the load on the network infrastructure making management and operations more complicated.

Optimizing both the network and the content delivery at the same time under multiple, some times even conflicting, constraints while guaranteeing the end-users expected quality of experience is a non-trivial, multi-dimensional optimization problem. Moreover, the market for content delivery as well as network providers is very competitive, leading both parties to investigate new ways to reduce capital investment and operating costs [40, 143].

## 1.2 Architectures, Trends and Opportunities

To address the challenges in content delivery, a variety of system designs have been proposed over the last decade. These solutions try to expand the CDI footprint by leveraging available resources of end-users or dynamically offloading the content delivery to other content delivery infrastructures e.g., in case of capacity bottlenecks or end-users in a network where the CDI has no close by servers. Figure 1.1 gives an overview of the various solutions and shows the level of involvement of each stakeholder in content delivery, namely the Content Producers (CP), Content Delivery Infrastructures (CDI), network operators (ISP), and the end-users. In this classification scheme the different roles are as follows:

**CPs** or *Content Producers* subsumes any type of business or private entity that has a primary interest (mainly financial) in end-users consuming its content. The content can either be created by the CP or licensed from others. Prominent examples of CPs are, e.g., news and infotainment sites, such as MSNBC or BBC, company websites like Volkswagen or Samsung, and software companies that digitally distribute their software and patches, such as Adobe or Microsoft. CPs that offer mainly third party licensed content include Online Social Networks (OSNs) like Facebook, Video on Demand (VoD) services like YouTube and Netflix. Recall, Netflix and Google.

**CDIs** or *Content Delivery Infrastructures* operate a dedicated infrastructure to distribute content of CPs to end-users. To offer reasonable performance CDIs need not only to operate enough infrastructure but also establish enough connectivity to the various networks that make up the Internet, be it by distributing servers into many networks or by connecting to them.

**ISPs** or *network operators* offer network and Internet access including but not limited to end-users and thus transport the content through their network. Well known ISPs are AT&T, Telefonica, or Deutsche Telekom.

Last but not least, **end-users** include everyone consuming content offered by CPs.

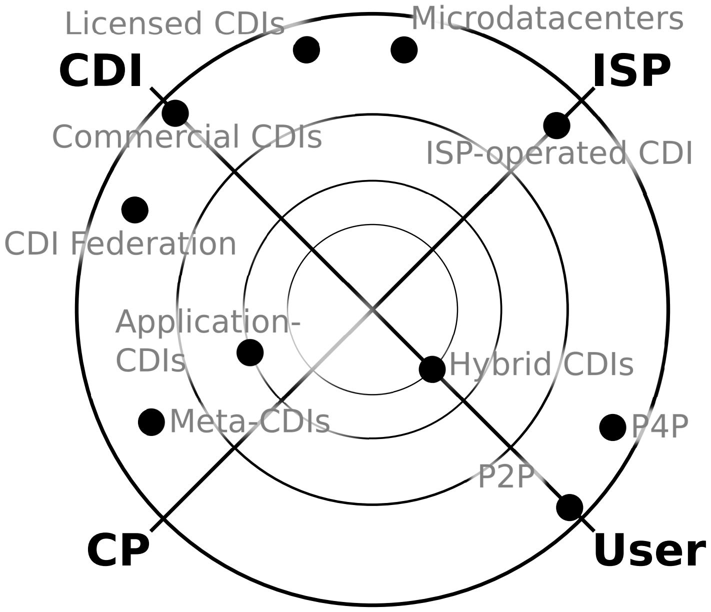


Figure 1.1: Content Delivery Spectrum

Note that in this classification an entity is not limited to a single role. For instance Google takes the dual role of Content Producer and Content Delivery Infrastructure with its YouTube service and in some places of the United States even has a third role by providing Internet access to end-users (e.g., Google Fiber in Kansas City).

### The Network Oracle

The classical approaches for content delivery are *commercial CDIs*, *ISP operated CDIs*, and *Peer-to-Peer Systems*. Commercial CDIs are independent business entities that operate large distributed server infrastructures to deliver content to end-users. They usually do not operate their own network infrastructure but instead rely on ISPs for network connectivity. ISP operated CDIs on the contrary do operate their own network but their server footprint is limited to the network footprint of the ISP. Peer-to-Peer systems are distributed architectures where the resources of the system are provided and operated by the end-users. In Figure 1.1 they are placed very close to their respective operators as the involvement of the other parties is marginal at best. For example, the ISP can throttle the P2P traffic of its customers to reduce the network utilization but this is more an indirect interaction with the content delivery itself. The same holds for peering or transit agreements with CDIs. Not the distribution itself is influenced but the traffic amount or delivery speed at which the distribution happens.

Earlier attempts to improve content delivery have been proposed in the area of P2P systems, which successfully utilize the aggregate capacity of end-users that are interested in downloading the same content [46]. Due to the popularity, openness, and availability of protocol specifications and client software, the research community was able to understand the drawback of such systems. The random connection to other peers (which increases the resilience of the system) in many popular P2P systems has put a high strain not only on the networks hosting the peers but also the connecting transit networks. As a result *P4P* [177] has been proposed as an ISP-P2P collaboration mechanism to better localize traffic. Augmented with network information, the peer selection can be improved and is able to avoid connection to peers in far away networks.

To utilize the systemic benefits of P2P systems and to scale up the infrastructure and at the same time reduce the capital investment in hardware, bandwidth and energy commercial CDIs [3] as well as ISPs [100] operate *hybrid content delivery Infrastructures* where end-users download content from the CDI servers as well as other end-users, mimicking the success of pure P2P systems. To avoid many of the complicated and time consuming contractual issues when deploying servers, commercial CDIs recently have started to offer their content delivery software to ISPs as *licensed CDI.* The administrative burdens to deploy, operate, and maintain servers inside their own network is much smaller for them and in some cases the licensed software is able to coordinate with the CDI operated servers forming a *CDI Federation.* The industries requirement for such an mode of operations has led to the CDNI working group [124] in the IETF which develops standards for necessary mechanisms and protocols. To allow Content Producers to take advantage of the many different CDIs and combining their individual strengths into a sort of virtual content delivery infrastructure *Meta CDIs* [59, 108] add an additional layer of abstraction to the process of content delivery. The Meta CDI selects for each end-user individually which of multiple available CDIs is used to deliver the desired content. This decision is based on multiple factors such as the network location of the user and the measured CDI performance among others and allows the Content Producer to influence the delivery process and at the same time improves the performance for the end-users.

So far all of the presented infrastructures and solutions were general purpose architectures. But some applications can benefit even more from an application specific optimization (examples are rate limiting for video streaming or server selection based on consistent hashing for very large files). This has led larger CPs to deploy *application specific CDIs* inside ISPs and highly connected data centers. Examples include Netflix Open Connect for video streaming [122] or Google Global Cache primarily for YouTube [26].

### The New Cloud

In the broadest sense todays Internet is an entanglement of “dump plumbing” to forward packets along paths and “highly integrated services” to provide additional in-network features such as caching, carrier-grade NAT, load balancing, or security features like intrusion detection or virus filtering. The launch of a new network service often requires another variety of proprietary hardware applicances and includes the increasingly difficult task to find the necessary space and power to accommodate these boxes. These difficulties and the need for a more *service centric network* has spurred another recent trend: marry cloud resources (processing and storage) with networking resources to meet the high performance requirements of bandwidth and storage critical applications such as high definition video streaming or delay sensitive applications like cloud gaming [153].

Improvements in virtualization technology and recent developments in network equipment architectures like Software Defined Networking (SDN) allows ISPs to migrate from proprietary hardware solutions to software based ones running on generic appliances deployed deep inside their network. While their initial intent often was to support only their own ISP specific services, such as ISP-operated CDNs, IPTV, carrier-grade NAT, deep packet inspection, etc., network operators now leverage these new capabilities to offer fully virtualized network and server resources in proximity to their end-users to third parties [54]. Major network operators around the globe, including AT&T, British Telekom, NTT, Deutsche Telekom and Telefonica, have recently joined their efforts to define the requirements for such a solution. Their draft, called *Network Functions Virtualisation* (NFV) [123], is currently in progress of standardization in the European Telecommunications Standards Institute (ETSI) [62]. The goal is to drastically reduce the complexity and number of different types of networking equipment by consolidating to an industry standard high volume server for fixed as well as mobile networks. A much anticipated side effect of such a solution is the avoidance of vendor lock-ins. These general purpose appliances, also called *microdatacenters*, are already deployed by large ISPs, including AT&T, Deutsche Telekom, and Telefonica, co-located with their major network aggregation locations (PoPs).

Other networking technologies, including SDN, aim to simplify network operations by decoupling the control plane from the data plane. SDN offers a logically centralized, programmable control of network traffic by introducing an abstraction layer of lower level functionality (e.g., forwarding data packets). Albeit reducing the dependency on vendor specific hardware, SDN nonetheless requires the network operators to replace their current networking equipment. The SDN approach is orthogonal and highly complementary to the introduction and deployment of NFV: either technology can be deployed independently from the other. In combination with cloud style computing, such as microdatacenters, SDN blurs the lines between networks and computing even further. The biggest advantage integrated network and cloud providers can offer is the ability to offer high quality cloud services as they control all resources on the path from the server to the end-user.

At the same time when the cloud started to move into the network, the research community started to leverage cloud resources to outsource most if not all of the network infrastructure (except the forwarding plane of course) and its control plane [17, 72, 157, 162]. By doing so network operators leverage the highly specialized knowledge in domain specific operations of the service provider to improve their own operations while reducing investment in up-to date technology and hardware and at the same time also reduce operational costs. Liu et al. argue in [107] for network providers to deploy ingress filtering to offer filtering of spoofed IP traffic to other networks as a service, not only to improve the efficiency of filtering spoofed IP traffic but also to create new revenue streams for the network operators at the same time. Sherry et al. show in [159] that it is not only possible to outsource nearly any network middlebox, such as firewalls, proxies, or even WAN optimizers, without impact on their performance but also to reduce their management complexity, cost, and capacity limits. Improving the situation even further, Olteanu et al. show that efficient migration of stateful middleboxes in cloud environments is feasible [129]. Kotronis et al. go even further and propose a system to completely outsource the routing control of a network to a third party service provider [94]. This enables the routing service provider to leverage a “bird’s eye view” on network clusters for making efficient routing decisions, detect and troubleshoot policy conflicts, and routing problems for improved efficiency and reduces operational cost. The ability to outsource network infrastructure enables ISPs to leverage economy of scale by deploying microdatacenters deep inside their network and utilize it for their own needs as well as capitalize on offering cloud resource close to the end-users to service providers, e.g., content delivery infrastructures.

So far our discussion about improving content delivery has touched the technical possibilities but neglected the incentives improving economics and market share. Both are key drivers towards collaboration which has been be observed in both the content delivery and the network operation business. On the one hand, large and already well established Content Delivery Infrastructures have a strong customer base among Content Providers and are responsible for delivering the content for their customers to the end-users around the world. Network operators on the other hand have a strong end-user base in their service region and are starting to offer cloud resources close to their end-users in aggregation locations (PoPs) of their network.

## 1.3 Problem Statement

Today’s content delivery landscape faces the problem of server allocation – where to place additional server resources – and user assignment – which end-user is assigned to which server. This is because CDIs are largely unaware of network conditions and end-user locations inside the ISPs network. However, this information has the potential to highly improve the efficiency and performance of allocating additional resources and assigning end-users to servers [66, 69, 137]. While some of this information can be inferred by measurements [128, 137] – a tedious and error prone task – a network operator has the information readily at hand! Therefore, we argue that collaboration between CDIs and ISPs is the next step in the natural evolution of deploying and operating content delivery infrastructures in the Internet.

## 1.4 Contributions

Despite the opportunities and benefits for collaboration, the mechanisms and systems to enable joint CDI deployment and operation inside the network are subject of this thesis. Therefore, we highlight the technical means leading to a win-win situation for all involved parties in content delivery. The contributions of this thesis are as follows:

### Content Delivery Landscape

First, the large spectrum of available content delivery architectures motivate us to investigate the current design and operating space of todays content delivery landscape and highlight the challenges content distribution faces. We find that the content delivery landscape is in a constant flux to further improve its delivery performance, increase its network footprint, and at the same time tries to reduce the capital investment and operational costs for its content delivery infrastructure. To quantify the potential benefits of a *collaborative* operation of content delivery infrastructures, we conduct a large scale measurement study of the largest commercial CDIs operations in an European Tier-1 ISP. We find that ample opportunities exist to leverage the ISPs knowledge about the current network state to enable better leverage the CDIs current infrastructure footprint.

### The New Cloud

Second, we identify two key enablers for collaboration in content delivery, namely informed user-server assignment and in-network server allocation. Until now, both problems have been tackled in a one-sided fashion by the CDIs. While informed user-server assignment improves the operation of already deployed content delivery infrastructures by taking network conditions, such as link utilization or number of backbone hops, into account, in-network server allocation offers an additional degree of freedom for the deployment of additional resources. It allows the CDI to freshly instantiate, migrate or shut down additional resources deep inside the ISPs network close to the end-users on short time scales, e.g., tens of minutes. Together the two enablers allow a joint optimization of network operations for mutual benefits and enables the deployment of new and highly demanding services and applications. This motivates us to propose a novel system design incorporating the two key enablers to improve content delivery through collaboration between CDIs and ISPs.

### NetPaaS

Third, we implement and evaluate a prototype system, called NetPaaS (Network Platform as a Service), realizing our design for collaborative server deployment and operation inside the ISP's network. We perform a first-of-its-kind evaluation based on traces from the largest commercial CDI and a large European Tier-1 ISP using NetPaaS. We report on the benefits for CDIs, ISPs, and end-users. Our results show that CDI-ISP collaboration leads to a win-win situation with regards to the deployment and operation of servers within the network, and significantly improves end-user performance. Our evaluation shows, that in the studied setting NetPaaS is able to reduce the overall network traffic by up to 7% and lower the utilization of the most congested link in the network by up to 60% when used solely for informed user-server assignment. When NetPaaS also offers in-network server allocation the delay for end-users is reduced significantly and up to 48% of all requests can be answered by a server located in the same PoP as the end-user with only 50 additional servers.

## 1.5 Outline

The rest of the thesis is structured as follows: Chapter 2 gives the necessary background information about protocols and technologies used in todays Internet content delivery. Chapter 3 consists of a survey of the current content delivery landscape, highlighting current and upcoming trends in its architectures and points out current challenges for the involved parties in content delivery. In Chapter 4 we conduct a measurement study of the largest CDIs in an European Tier-1 provider highlighting the opportunities for collaboration to improve content delivery. We identify and formalize two key enablers, namely informed user-server assignment and in-network server allocation, for collaboration between CDIs and network operators in Chapter 5. In Chapter 6 we propose a novel system architecture, called NetPaaS, leveraging the two key enablers to improve content delivery in the Internet. We also discuss the scalability and privacy related issues of the system and how said system can be integrated into todays operation of Content Delivery Infrastructures. Chapter 7 evaluates NetPaaS using operational data from the biggest commercial CDI and an European Tier-1 network provider. We show that joint server deployment between CDIs and ISPs can improve content delivery significantly in the studied setting.

# 2
# Background

In this chapter we review the basic building blocks required to understand todays landscape of content delivery infrastructures. We start by introducing the *Internet Service Provider (ISP)* as the managing entities of the Internet and continue our excursion with the introduction of the two most important protocols in content delivery today, namely the *Domain Name System (DNS)* protocol and the *HyperText Transfer Protocol (HTTP)*. We then explain how content delivery works using a short example and describe the general architecture and all relevant components of a *Content Delivery Infrastructure (CDI)* and. Next, we provide a short overview of *Virtualization* techniques, as they offer unprecedented flexibility in resource allocation and management and are an essential component of recent large scale infrastructure deployments, such as cloud computing. Last but not least, introduce and shortly discuss the *Peer-to-Peer (P2P)* paradigm for content delivery.

## 2.1 The Internet & You: Internet Service Providers

The Internet is a world wide network of networks with the infrastructure of those networks provided by *Internet Service Providers (ISPs)*. Generally speaking, an ISP is a business or organization that operates a dedicated network infrastructure and offers Internet access to its customers. The interconnection of multiple individual networks run by ISPs forms what we commonly call the Internet. The general layout is shown in Figure 2.1: End-users and customer networks (e.g., corporate networks) obtain connectivity from ISPs which in turn are interconnected, either directly or

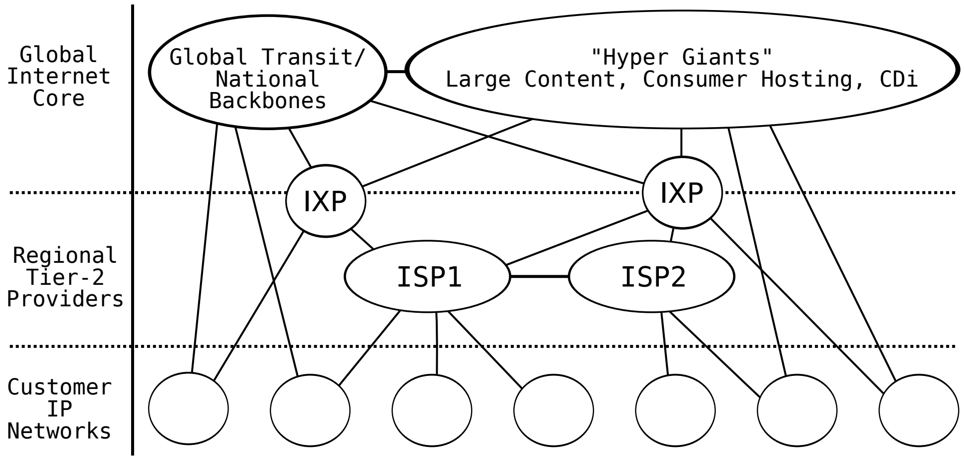


Figure 2.1: Layout of the Internet Structure [98]

through national transit or global backbone providers[1]. In addition, the Internet in the last decade has experienced the ascent of a new type of network, the so called *Hypergiants*. Hypergiants are large networks that mainly host content that end-users are interested in, such as Google and Netflix. They usually generate huge amounts of traffic and thus thrive to directly interconnect with ISPs.

The layout shown in Figure 2.1 also highlights the clear distinction between the individual networks run by ISPs and the Internet: the administrative control over the individual network infrastructures remains solely with the ISPs. This also implies that no single entity can coerce control over the Internet as each ISP controls only its own network and the direct connections to other networks.

The customers of ISP can be, e.g., end-users, hosting facilities, or even other networks. End-users can be connected via a wide range of access technologies, such as dial-up-modems, digital subscriber line (DSL), fiber to the home (FTTH) or wireless technologies such as 3G, WiMax, or satellite links. If the ISP offers access to end-users via one or more of such technologies, it is also called an “access ISP”. If other networks use the ISP to reach another network, the ISP is called a “transit ISP”, as the traffic crosses the ISPs network but neither originates nor terminates in the ISPs network. When the ISP offers other networks connectivity to Internet, that is it allows them to send traffic to the Internet via its own network, the ISP is called an “upstream ISP”. Note that an ISP can have multiple roles at the same time, e.g., a large access ISP can also offer transit for other networks.

To be able to interconnect with other networks an ISP needs to operate an autonomous system (AS). An AS is an administrating entity, generally under the control of one administrative domain, for one or more publicly routable IP prefixes and

[1] Transit and backbone operators are basically large network operators with a national or global footprint that offer connectivity to ISPs just like they offer connectivity to their customers.

requires an officially assigned and unique autonomous system number (ASN). Both the ASNs and publicly routable IP prefixes are governed by the Internet Assigned Numbers Authority (IANA) which delegates the assignment to the Regional Internet Registires (RIR). Each AS is usually managed by an Interior Gateway Protocol (IGP), e.g., OSPF [120] or ISIS [131]. Since an AS is run centrally by one instance, there is no need for information aggregation and/or hiding.

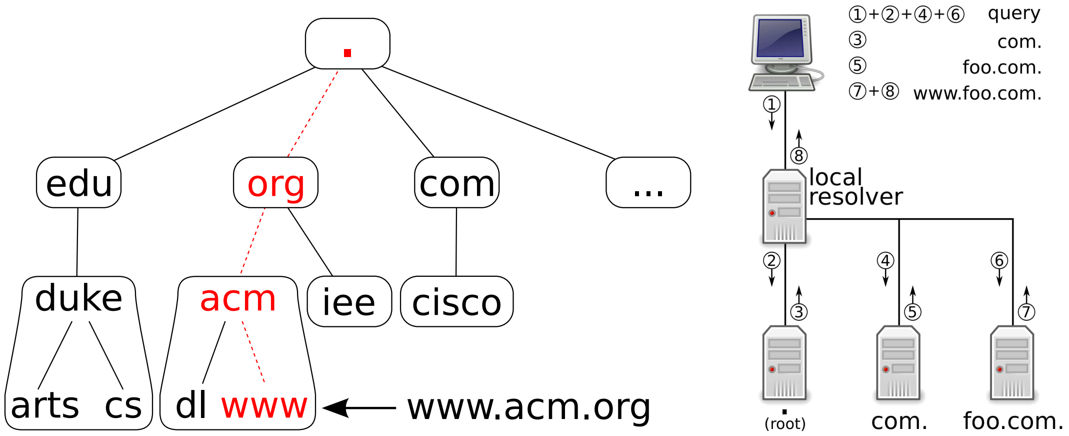
To interconnect different ASes the Border Gateway Protocol (BGP [147]) is the de-facto standard used and provides the required IP prefix reachability information to make routing decisions in the Internet. To keep the distribution of routing information scalable throughout the Internet, the entire internal management of the individual AS is abstracted and aggregated. Each AS announces which IP prefixes can be reached via its network and other networks use this information to make routing decision, that is which network path they use to send traffic along towards its destination. For example in the case of an upstream ISP, the ISP announces all IP prefixes it knows to its customers, while the customers would only announce their own public IP prefixes to the ISP. When an AS needs to communicate with another AS that it does not have a direct connection to, the communication has to transit one or more different ASes. Thus, along with with the pure reachability information, the ASN is also transmitted. This allows for loop detection as well as an estimate of how many AS hops away a destination is.

The greatest challenge for an ISP is the efficient operation of its infrastructure. To this end, ISPs usually apply a process called *Traffic Engineering (TE).* TE is, simply speaking, the process of adjusting the internal routing weights and BGP announcements such that the traffic flows through the network in the most effective way. This is usually done to avoid link congestion and reduce delays by using short paths, but also to reduce the capital expenses by reducing the utilization of expensive peering links.

## 2.2 Domain Name System

Before 1983, a simple plain text file (*hosts.txt*) was used to translate hostnames into IP addresses. Back then, it was manually distributed to all hosts connected to the Internet. With a growing number of hosts scalability and management issues became more and more rampant. To alleviate them, the *Domain Name System (DNS)* [118] was introduced in 1983 and has been a key part of the Internet ever since.

DNS is a distributed database with a hierarchical structure and divides the complete Internet namespace into *domains.* As "Naming follows organizational boundaries, not physical networks" [118,167] the administration of domains is organized in *zones.* This information is distributed using *authoritative name servers.* The top most level of the DNS hierarchy starts with the *root zone* using 13 globally distributed and replicated *root name servers.* To mark the boundary between hierarchy levels in



(a) Partial DNS name space with zones (circled).

(b) Hostname lookup.

domain names the "." character is used. The root zone has an empty domain label and therefore is represented by a dot. Responsibility for specific parts of a zone can be delegated to other authoritative name servers which can in turn delegate responsibilty further. For example, the root zone delegates responsibility for, e.g., the `.org` domain to the Public Interest Registry which in turn delegates responsibility for `acm.org` to the Association for Computing Machinery (ACM). The information regarding a particular domain of a zone is stored in *Resource Records (RRs)* which specify the class and type of the record as well as the data describing it. To improve scalability and performance, DNS heavily relies on caching. The time for which a specific RR can be cached is determined by its *Time To Live (TTL)* and is part of the zone configuration. In the end, each domain is responsible for maintaining its own zone information and operates its own authoritative name server. An alternative view of the domain name space is a tree with nodes containing domain labels separated by dots. Figure 2.2a illustrates this view of the partial domain name hierarchy including the administrative organization into zones.

To resolve a domain name, the end-hosts *stub resolver* usually queries a local name server called *caching resolver*. If the information is not available in the resolvers cache, it queries the authoritative name server of the domain. In case the resolver does not know how to contact the server, it queries a root name server instead. The root name server *refers* the resolver to the authoritative name server responsible for the domain directly below the root. This referrals continue until the resolver steps down the domain name space tree from the root to the desired zone and is able to resolve the domain. In our example, the caching resolver is called an *iterative* resolver, as it iteratively queries the authoritative name servers until it can resolve the hostname, while the end-hosts stub resolver is called a *recursive* resolver, as it leaves the hostname resolution completely up to the caching resolver. Figure 2.2b illustrates recursive (steps 1 & 8) and iterative (steps 2 - 7 ) hostname resolution.

Today, DNS plays a major role in content delivery [19, 37, 117, 128], especially for assigning end-users to CDI servers. Low TTLs enable CDIs to quickly react to demand surges and allows fine grained load balancing. Crafting DNS replies based on the querying caching resolvers geo-location results in short delays and traffic localization. However, such practices have attracted criticism [6, 172] largely due to reduced cacheability and increased network load because of low TTLs. Furthermore, the basic assumption that end-users are generally close to the used caching resolver does not always hold true [6].

## 2.3 HyperText Transport Protocol

The *Hypertext Transfer Protocol (HTTP)* [63] has become todays de-facto standard to transport content in the Internet [43, 71, 98, 137, 151]. Introduced in 1989 by Tim Berners-Lee at CERN (Conseil Européen pour la Recherche Nucléaire) and published in 1991 as version HTTP/0.9 by the *World Wide Web Consortium (W3C)* [31] and standardized by the *Internet Engineering Task Force (IETF)* in several *Requests for Comments (RFCs)* [32, 63, 89, 126] defining HTTP as an "application-level protocol for distributed, collaborative, hypermedia information systems". The version that is today in common use is HTTP/1.1. The upcoming standard HTTP/2.0 is currently under development in the *HTTPbis* working group [125].

HTTP is a simple plain-text *request-response* protocol on top of TCP/IP[2] and follows a *client-server* architecture. It allows end-users to request, modify, add or delete resources identified by *Uniform Resource Identifiers (URIs)* – or *Unified Resource Locators (URLs)*, but today both are used as synonyms [116]. A valid URI consists of three parts: the protocol schema (e.g., *http://* for HTTP), the domain name (such as *www.example.com*, but a literal IP address is also possible) and the full path to the resource (for example */path/to/resource*. The resulting URI from our example would be: `http://www.example.com/path/to/resource`. The type of the resource often corresponds to a file but can also be dynamically assembled content or the output of an executable on the Web server.

Every HTTP message consists of an introductory line, optional *header lines* specifying additional information and a potentially empty message body carrying the actual data. The introductory line of a HTTP request, see Listing 2.1 (left), consists of a *method* and the *URI* it should act upon. Similarly, the introductory line of a reply, see Listing 2.1 (right), contains a standardized three-digit status code and a textual representation specifying if the request was successful or not. Although primarily designed for the use in the Web, HTTP supports more operations than fetching a Web page. For a full list of available methods and status codes in HTTP/1.1 and their description, see Table A.1 and Table A.2 in the appendix. Both request and

[2] Although the RFC mentions the possibility to use UDP as well it is not widely used today.

```
GET / HTTP/1.1                  HTTP/1.1 200 OK
Host: www.example.com           Accept-Ranges: bytes
User-Agent: Mozilla/5.0 [...]   Content-Type: text/html; charset=UTF-8
Accept: text/html [...]         Date: Mon, 29 Jul 2013 15:46:02 GMT
Accept-Language: en-US,en;q=0.5 ETag: "780602-4f6-4db31b2978ec0"
Accept-Encoding: gzip, deflate  Last-Modified: Thu, 25 Apr 2013 16:13:23 GMT
Connection: keep-alive          Server: ECS (iad/1984)
                                X-Cache: HIT
                                Content-Length: 1270

                                <!doctype html>
                                <html>
                                [...]
```

Listing 2.1: HTTP request (left) and response (right) for www.example.com

reply messages may be followed by one or more *header lines*, see lines 2–9 in Listing 2.1, specifying additional information, e.g., the character set the client accepts or for how long a client may cache the response. Some headers are only valid in requests, others only in replies and some are valid in either direction. For a list of standardized HTTP headers, see e.g., [63, 127].

To improve performance and efficiency HTTP has built-in support for *caching* of content. The *Expires* header tells a client for how long a response can be considered valid and thus loaded from the local cache. Yet, not all answers come with an expires header, what makes caching non trivial. Therefore, HTTP supports a *conditional GET* where the server transmits the object only if it has changed since it was transferred to the client. For this the client can use, e.g., the Last-Modified (see line 6 of the HTTP reply in Listing 2.1) or If-Modified-Since headers in the request.

Another important mechanism supported by HTTP is *redirection.* The 3xx status codes allows an Web server to redirect individual users to other servers, e.g., if the Web server is under high load or another Web server is closer to the client. However, the drawback of redirection is the additional delay due to having to open another TCP connection to the new Web server.

Although HTTP in itself is a stateless protocol – that is the server does not need to keep state between successive requests from the same client – technologies such as session parameters or HTTP cookies enable Web sites to keep state. In both cases the state is stored on the client side and is transferred to the Web server with each request. Session parameters are simply key value pairs that can be attached to the URI. Cookies are small pieces of data stored on the end-users computer by websites. Such state information is usually required by dynamic content, such as personalized Web pages, or for authentication purposes.

The most recent version HTTP/1.1 includes some changes to improve the overall performance of the protocol. While HTTP/1.0 did close the underlying TCP connection after it received the requested resource, HTTP/1.1 supports persistent connections, sometimes also called HTTP keep-alive or HTTP connection reuse. It allows a client to receive multiple resources over a single TCP connection by sending a new request after the response to the previous request. This avoids additional delay cause by the necessary TCP 3-way handshake and bandwidth limitations due to the slow start phase of newly created TCP connections. The HTTP connection in Listing 2.1 uses this feature, see line 7. In addition to that, HTTP/1.1 supports pipelining, that is, multiple resources can be requested by the client without waiting for the respective responses from the Web server which greatly reduces the time to load multiple resources especially on high delay connections, such as satellite links.

HTTP/2.0 is expected to substantially improve end-user perceived latency through asynchronous connection multiplexing, header compression, and request-response pipelining. Therefore, it does not require multiple TCP connections to leverage parallelism and thus improves the use of TCP, especially regarding TCP's congestion control mechanisms. HTTP/2.0 retains the semantics of HTTP/1.1 and therefore leverage existing standardization on HTTP methods, status codes, URIs, and where appropriate, header fields. For more information, see [125].

## 2.4 Content Delivery Infrastructures

Over the past decades the demand for content has seen phenomenal growth and is still expected to grow [43]. In addition, many already and newly deployed services gain additional benefits from improved performance, e.g., reduced latency, in content delivery [91]. The need for increased capacity and improved performance has led to the emergence of *Content Delivery Infrastructures (CDIs)*: large dedicated infrastructures to deliver content to end-users around the world. Traditionally, content is placed first on the Web servers of the *Content Producer (CP)*, the original Web servers. Content delivery infrastructures are specifically designed to reduce the load on the origin servers and at to improve the performance of end-users.

In general, there are three main components in a CDI architecture: a server deployment, a content replication strategy and a mechanism for directing users to servers. But not all CDIs are built upon the same philosophy, design, and technology.

The server deployment strategy is one of the most crucial factors in any CDI architecture and has a high influence on the possible performance gains. Therefore, we dedicate a full chapter to the CDI deployment strategies: Chapter 3 gives a detailed overview of the current content delivery landscape and we discuss the challenges content delivery faces today in Chapter 3.1. The classical deployment strategies for content delivery infrastructures are described in Chapter 3.2 and in Chapter 3.3 we introduce emerging trends in content delivery, such as Hybrid and Meta CDIs.

In the remainder of this section we want to introduce the different possible solutions for the three main components of a CDI and discuss their various benefits and drawbacks. To introduce the general concept of content distribution, Chapter 2.4.1 provides an illustrative example of how content delivery using CDI resources works in general. Chapter 2.4.2 introduces the two main concepts for content replication: push based and pull based content replication. In Chapter 2.4.3, we will introduce the different mechanisms to assign end-users to CDI server and discuss their benefits and drawbacks. Remember that the detailed discussion on the different deployment strategies is left for the next chapter.

### 2.4.1 Content Delivery 101

The goal of this section is to introduce the general concept of content delivery in the Internet. Figure 2.3 shows an example of how content delivery infrastructures are embedded into the Internet architecture and how the resulting traffic flows to the end-users look like. Recall, the Internet is a global system of interconnected Autonomous Systems (ASes), each operated by an Internet Service Provider (ISP), see Chapter 2.1. The example shows three ASes, numbered 1–3, with each AS operating a couple of backbone routers. For inter-connectivity, AS1 has established a peering link with AS2 and AS3 while AS2 and AS3 have established two peering links. A Content Producer (CP), **example.com**, utilizes a centralized hosting infrastructure in AS2 to deliver the HTML Web page depicted in Figure 2.4. The Web page also contains two images, *img1.png* and *img2.png*, that are distributed by two different CDIs, **cdi-a.com** and **cdi-b.com**.

The server location differs from CDI to CDI and depend on contractual agreements between the CDI and the individual ISPs. In some cases the servers are deployed in the data centers of the ISP or deep within the network, e.g., co-located in the network aggregation points (PoPs), and therefore belong to the same AS. End-users of those ISPs are typically served by the CDI servers inside the ISPs network. The first CDI, **cdi-a.com** utilizes such an approach and has deployed its servers deep inside the network of AS1, location $\alpha$, and AS3, location $\beta$. In other cases CDIs utilize multiple well connected datacenters with direct peerings to ISPs. The second CDI, **cdi-b.com**, utilizes this approach and has servers deployed in two datacenters to deliver content to the end-users. Datacenter I has a direct peering with AS1 while datacenter II is multihomed[3] with connectivity to AS1 and AS3. With other ISPs there may be no relationship with the CDI at all and the traffic to the end-users of those ISPs is routed via another AS, the so called transit AS.

Let us consider the steps that are necessary to download the Web page shown in Figure 2.4. This page consists of the main HTML page *index.html* located at `http://www.example.com/index.html` and two embedded image objects, *img1.png* and

[3] Multihoming describes the fact that the datacenter is connected to more than one network providing Internet access.

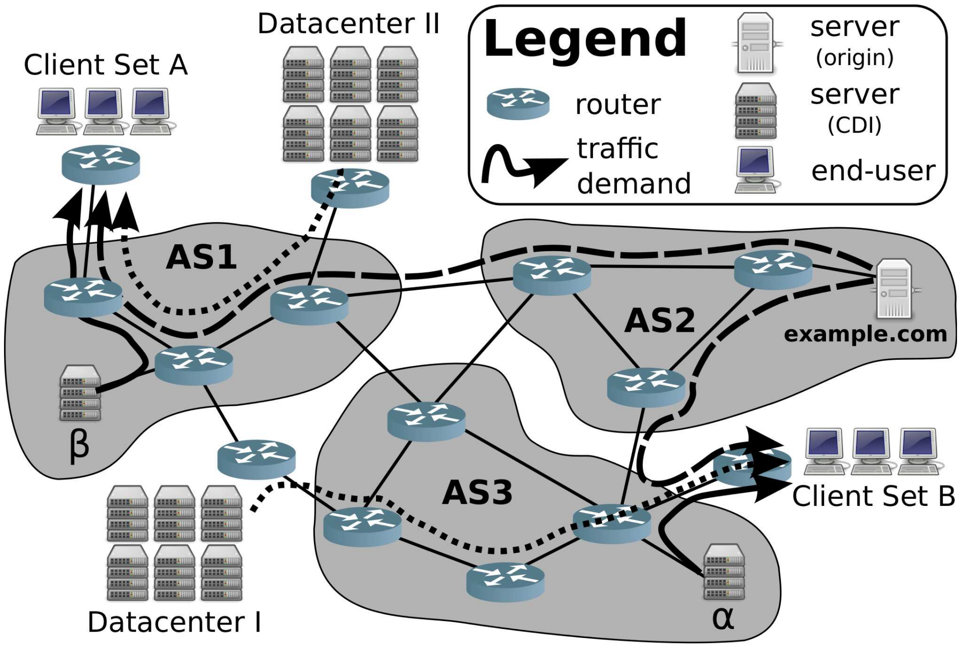


Figure 2.3: Example of CDI deployments and traffic flows (Web traffic demands).

*img2.png* located at `http://cdi-a.com/img1.png` and `http://cdi-b.com/img2.png` respectively. The Content Producer responsible for **example.com** has decided to use the services of two CDIs to deliver the embedded images, while the main HTML page (*index.html*) is served from the CPs own centralized hosting infrastructure in AS2. The first image (*img1.png*) is hosted by **cdi-a.com** and the second image (*img2.png*) by **cdi-b.com**. The resulting traffic flows are shown in Figure 2.3.

If a specific client from client set A in AS1 requests the Web page at `http://www.example.com/index.html` it first resolves the hostname *www.example.com* using the Domain Name System (DNS) which returns the IP address of a server from the centralized hosting infrastructure of the CP in AS2. The client then utilizes the HTTP protocol to connect to the Web server and requests the HTML page *index.html*. After receiving the Web page the client needs to get the two embedded image objects to be able to render the full Web page. It will again resolve the hostnames using DNS and the CDIs in question will return the IP address of the "nearest" server based on the clients location. In the case of our client from set A, **cdi-a.com** will utilize a server from location $\alpha$ in AS1 to deliver *img1.png*, while **cdi-b.com** uses datacenter I to serve the second image object *img2.png*. In contrast, if a specific client from client set B requests the Web page, the two image objects hosted on the CDI infrastructure are delivered from different servers, namely a server in location $\beta$ for **cdi-a.com** and another server in datacenter II for **cdi-b.com** respectively. The main HTML page *index.html* on the other hand is still delivered from the centralized hosting infrastructure of the CP in AS2. The resulting traffic

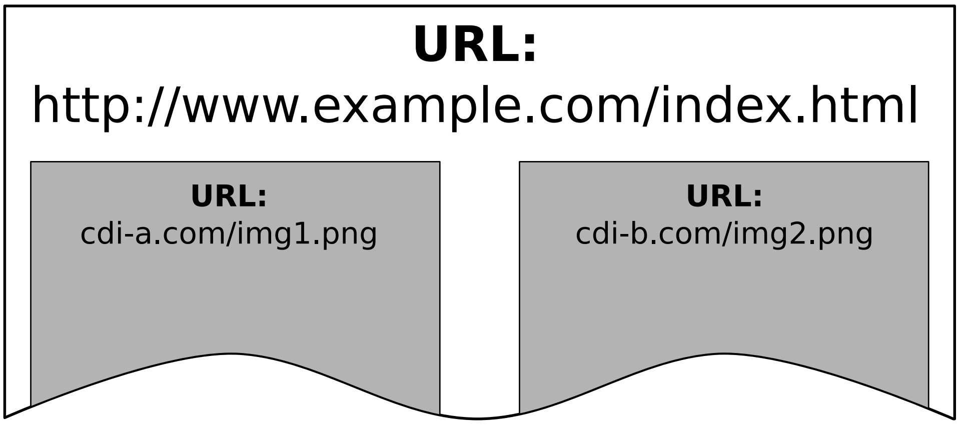


Figure 2.4: Example Web page with some CDN content.

flows are depicted in Figure 2.4, which also shows the advantage of utilizing CDIs to deliver content, namely the shorter distance between the end-user and the server delivering the content and to some extend the avoidance of inter-AS peering links.

### 2.4.2 Content Replication

Content replication in the context of content delivery infrastructures describes the process of duplication and distribution of content from the origin Web server to the CDI servers which store the content locally for fast access. This enables the CDI server to satisfy requests for content directly from the local storage, the so called cache, without the need to fetch it from the origin Web server first. An important aspect of content replication is the coherence of the content in the local cache and the origin Web server. The content replication mechanism in place must ensure that the content stored in and served from the local cache is the same as if served from the origin web server. Highly related to the content replication mechanism is the caching algorithm which is used to determine which objects are stored, updated or evicted. There is an entire field of research dedicated to this area and thus out of scope for this thesis. For more information see, e.g., [1, 28, 52, 136, 173].

A very simple form of content replication implies having a local copy of all objects from the origin Web server. But the tremendous amount of content with frequent additions and updates in combination with the huge number of servers that constitute todays content delivery infrastructures make this approach technically and economically infeasible. So far mainly two different content replication strategies are employed in content delivery today:

In **pull based** content replication a request for content that is not available in the local cache will trigger a recursive request at the CDI server. When a requested object is not locally available the server will first try to fetch it from neighboring servers in the same cluster or region. If the object is not available at neighboring servers, the origin server responsible for the object is contacted to retrieve the object. The received object is first stored in the local cache and then delivered to the end-user. To keep the content up to date, objects are usually assigned a time-to-live (TTL) value, which describes for how long this copy can be considered valid. If the TTL of an object is no longer valid it can be re-fetched or evicted from the cache. The pull based content replication strategy allows the CDI to assign any user to any cache as it ensures that the content, if not locally available, will be fetched from the origin server and then served to the end-user. This increases the scalability of the content delivery infrastructure [169] and is used by many CDIs today [128]. Yet, a slight drawback exists, the first request for each object will result in a cache miss and the resulting recursive request will induce an increased delay for the end-user that issues the original request. Also the limited local storage might result in objects being evicted from the cache and thus again create cache misses and increased delays.

**Push based replication** describes the approach where content is duplicated and actively distributed or "pushed" to some or all CDI servers. This strategy tries to avoid the inital cache miss that is inherent in the pull based content replication approach and allows the CDI to pre-populate the servers before the demand for content is expected to begin. This scenario is especially interesting for large scale events that can be planned in advance, e.g., airing a new episode of a popular TV series. Moreover, it alleviates the need for a caching algorithm as the required local storage is known in advance. In contast to the pull based content replication approach, the push based approach does not allow the CDI to assign end-users to arbitrary servers but requires the decision to consider the locally stored objects on each server of the content delivery infrastructure. Considering the huge number of servers of todays content delivery infrastrucures and the tremendous amount of storage (and thus objects) modern servers have, this is by no means an easy task. As a result, the complexity of this approach and thus the whole content delivery system is increased manyfold. Moreover, every mistake, even when caused by e.g., faulty or mis-behaving middleboxes, in the server assignment will result in object or (even wose) page load errors deminishing the end-users quality of experience significantly. However, combined with pull based content replication, this approach is actively used, especially by CDIs delivering large objects, e.g., high definition video or software.

### 2.4.3 End-User to Server Assignment

To complete the picture one question remains. How does the CDN choose the "nearest" server to deliver the content from? Today's CDN landscape relies mainly on three techniques to assign end-users to servers.

1. IP-Anycast
2. DNS based redirection
3. HTTP redirection

While all techniques help the CDNs to assign end-users to their servers, all of them have different drawbacks, the most notable being the possible inaccuracy due to end-user mis-location. Chapter 3.1 will provide more details on this and other challenges content delivery faces today and Chapter 5.3.1 presents various solutions to overcome some of them. The remainder of this section will explain how the different techniques for assigning end-users to CDI servers work and also shortly discusses their limitations:

**IP-Anycast:** IP Anycast is a routing technique used to send IP packets to the topologically closest member of a group of potential CDN servers. IP Anycast is usually realized by announcing the destination address from multiple locations in a network or on the Internet. Since the same IP address is available at multiple locations, the routing process selects the shortest route for the destination according to its configuration. Simply speaking, each router in a network selects one of the locations the Anycasted IP is announced from based on the used routing metrics (e.g., path length or routing weights) and configures a route towards it. Note that, if a network learns of an Anycasted IP address from different sources, it does not necessarily direct all its traffic to one of its locations. Its routing can decide to send packets from region A in the network to location A' while region B gets a route to location B'. This means that the entire server selection of a CDN becomes trivial as it is now a part of the routing process. This means that the CDN loses control of how the users are mapped to the server because the network calculates the routing based on its own metrics. Another issue is that the routing in a network is optimized based on the ISPs criteria which might not be the same as the CDNs or even contrary. Thus the “nearest” server might not be the best one the CDN could offer.

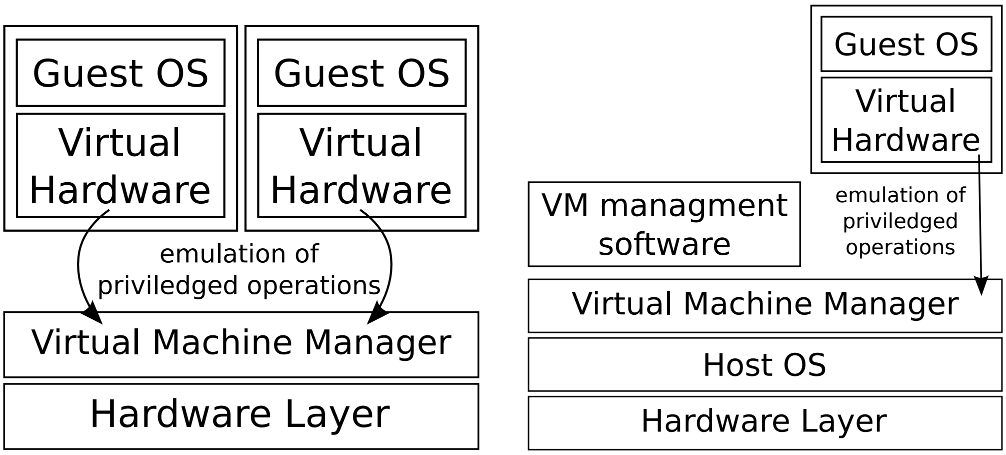
**DNS based redirection:** Today most CDNs rely on the Domain Name System (DNS) to direct users to appropriate servers. When requesting content, the end user typically asks a DNS resolver, e.g., the resolver of its ISP, for the resolution of a domain name. The resolver then asks the authoritative server for the domain. This can be the CDN's authoritative server, or the the content provider's authoritative server, which then delegates to the CDN's authoritative server. At this point the CDN selects the server for this request based on where the request comes from. But the request does not come directly from the end-user but from its DNS resolver! Thus, the CDN can only select a server based on the IP address of the end user's DNS resolver. To improve the mapping of end users to servers, the client-IP eDNS extension [48] has been recently proposed. Criteria for server selection include the availability of the server, the proximity of the server to the resolver, and the monetary cost of delivering the content. For proximity estimations the CDNs rely heavily on network measurements [128] and geolocation information [114] to figure out which of their servers is close by and has the best network path performance. A recent study [6] showed that sometimes the end user is not close to the resolver and another study points out that geolocation databases can not be relied upon [141]. Thus the proximity estimations for the “nearest” CDN server highly depend on the quality and precision of network measurements and a proper DNS deployment of the ISPs.

**HTTP redirection:** The Hypertext Transfer Protocol (HTTP) is today's de-facto standard to transport content in the Internet (see Chapter 4.1.1). The protocol incorporates a mechanism to redirect users at the application level at least since it was standardized as version 1.0 in 1996 [32]. By sending an appropriate HTTP status code (HTTP status codes 3xx, see Chapter 2.3) the web server can tell the connected user that a requested object is available from another URL, which can also point to another server. This allows a CDN to redirect an end-user to another server. Reasons for this might include limited server capacities, poor transfer performance or when another server is closer to the end-user, e.g., a client from the US connecting to a server in Europe although the CDN has servers in the US. The HTTP redirection mechanism has some important benefits over the DNS based approach. First, the CDN directly communicates with the end-user and thus knows the exact destination it sends the traffic to (opposed to the assumption that the DNS resolver is "close"). Yet, it still has to estimate the proximity of the end-user using the same methodologies as described in the DNS based case. Second, the CDN already knows which object the end-user requests and can use this information for its decision. It allows a CDN to direct a user towards a server where the content object is already available to improve its cache hit rate. Other important informations includes the size and type of the object. This allows the CDN to optimize the server selection based on the requirements to transfer the object, e.g., for delay sensitive ones like streaming video or more throughput oriented ones like huge software patches. Yet, this improvement comes at a price as the user has to establish a new connection to another server. This includes another DNS lookup to get the servers IP address as well as the whole TCP setup including performance critical phases like slow start. This can repeat itself multiple times before an appropriate server is found, which delays the object delivery even further.

## 2.5 Virtualization

In recent years, *virtualization* has revolutionized the way we build systems [135]. Major advances in performance, stability and management and the availability of off-the-shelf solutions has led to what we today know as "The Cloud": dynamic allocation of virtually unlimited resources on demand. This new deployment paradigm becomes more and more important for any large scale system and therfore is a highly relevant aspect for content delivery architectures.

In 1960 IBM developed virtualization originally as a means to partition its large mainframe computers into several logical units. The capability of partitioning available resource allowed multiple processes to run at the same time, thus improving efficiency while at the same time reducing maintenance overhead. Remember, back in this time computers were only capable of running a single process and batch-processing was considered state of the art in computer science. None of the very



(a) Type 1 Hypervisor.

(b) Type 2 Hypervisor.

Figure 2.5: Full Virtualization.

basic operating system (OS) technologies we have today, such as interrupts, process, or memory management, were existing back in the 60s [167].

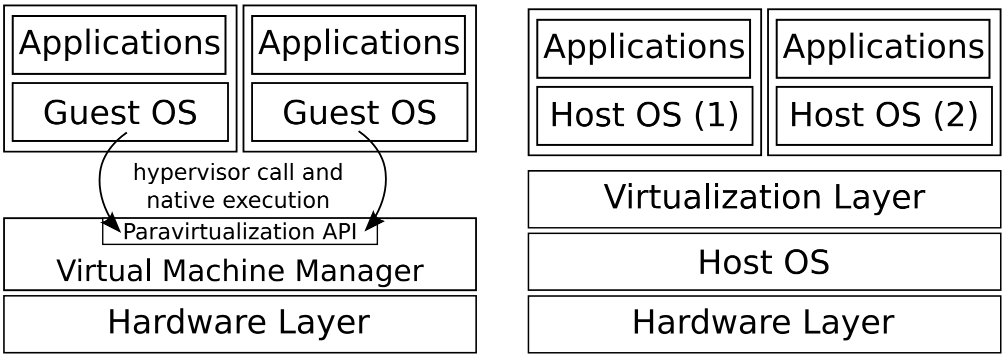
Today *virtualization* is commonly defined as a technology that introduces an intermediate abstraction layer between the underlying hardware to the operating system (OS) running on top of it. This abstraction layer, usually called *virtual machine monitor (VMM)* or *hypervisor*, basically conceals the underlying bare hardware and instead presents exact virtual replicas to the next layer up. This allows the hypervisor to partition the available hardware into one or more logical units called *virtual machines (VMs)* and thus to run multiple and possibly different OSes in parallel on the same physical hardware [150, 167].

The benefits of virtualization are manifold and include:

- **Failure Mitigation**: A failure in one VM does not influence the other VMs.
- **Consolidation**: Fewer physical machines take up less space and power and require less capital investment in hardware.
- **Management**: VMs can be easily allocated, de-allocated or migrated, and the virtual hardware can dynamically adjusted to fit changing requirements.
- **Strong Isolation**: VMs are completely isolated from each other and a compromised VM does not result in all VMs being compromised.

Today, multiple different approaches of virtualization exists and we next explain them in more detail.

**Full Virtualization:** The *Full Virtualization* [150] approach completely virtualizes the underlying hardware and exposes exact replicas to the OS(es) running on top. It comes in two different flavors that differ on what the VMM or hypervisor runs on. In the first type, usually referred to as *type 1 hypervisor*, runs directly on top of the bare



(a) Paravirtualization.

(b) OS Level Virtualization.

Figure 2.6: Para- and OS-Virtualization.

metal hardware, see Figure 2.5a. In reality, the type 1 hypervisor can be considered an OS as only the hypervisor can execute *privileged* instructions on the CPU, while the VMs privileged instructions causes a trap to turn control to the hypervisor. The hypervisor will inspect the privileged instruction and emulate the exact behavior of the real hardware. However, to enable this virtualization approach, the CPU needs to support traps for privileged instructions of VMs running in non-privileged mode. To enable virtualization on CPUs without support for privilege traps *type 2 hypervisors* run on top of an existing host OS, usually as a normal user-space application, see Figure 2.5b. They also provide virtual replicas of the hardware to the virtual machines and manage the access to the physical hardware by fully emulating its behavior in software. In addition, all privileged instructions are replaced by calls to a function that handles the instruction in the hypervisor, a technique known as *binary translation*. Although one might expect that type 1 hypervisors greatly outperform type 2 ones, this is not the case. The reason for this is that traps require CPU context switches and thus invalidate various caches and branch prediction tables. Type 2 hypervisors replace the corresponding instructions with function calls within the executing process and thus do not incur the context switching overhead. The best known virtualization solutions using the Full Virtualization approach are VMware Workstation [148] and Oracle VirtualBox [130].

**Paravirtualization:** To further improve the performance of VMs, the *Paravirtualization* [167] approach requires modifications to the guest OS to make *hypervisor calls* instead of executing privileged operations, similar to processes making system calls in the OS. To this end, the hypervisor exposes an API (Application Programming Interface) that alleviates the need to emulate peculiar hardware instructions and exact semantics of complicated instructions by shifting the execution of privileged instructions to the hypervisor. This results in a significant performance gain for the guest OS. Prominent examples of the Paravirtualization approach, see Figure 2.6a for an illustration, are Xen [30] and VMWare ESX(i) [148].

**OS Level Virtualization:** In *OS Level Virtualization* [150] the hardware is virtualized at the OS level, where the "guest" OS shares the environment with the OS running on the hardware, i.e., the running host OS kernel is used to implement the different "guest" environments. Applications running in the guest environments see them as dedicated and isolated OSes. The main advantage of this approach, see Figure 2.6b, is the simplicity of its implementation and almost no performance impact on the application. On the downside, the VMs are limited to the kernel and system environment of the host OS and a compromised VM endangers all other VMs as well, as the attacker gains access to the host. Prominent examples of this virtualization approach are BSD Jails [85], Solaris Containers [99] and Linux VServers [53].

**Virtualization Today:** Virtualization technology has revolutionized the way systems are built [135] and has seen major advances in terms of performance and stability. Once a major concern, the overhead of virtualization is negligible today, with VM boot up times in the order of tens of seconds and almost no runtime overhead [150, 175]. In addition, many tools for VM management have been developed and today a number of off-the-shelf solutions to spin a virtual server based on detailed requirements [110] are readily available from vendors such as NetApp and Dell. Over the past years, big Cloud providers [19, 117] have deployed large virtualized server infrastructures to leverage economy of scale and consolidation potentials to offer VMs as Infrastructure as a Service (IaaS) to its customers. We conclude that virtualization is a mature technology offering flexible ways of deploying and managing server infrastructure on a large scale.

## 2.6 Peer-to-Peer Networks

The P2P paradigm has been very successful in delivering content to end-users. BitTorrent [46] is the prime example, used mainly for file sharing and synchronization large amounts of data. Other examples include more delay sensitive applications such as video streaming [59, 97, 109]. Despite the varying and perhaps declining share of P2P traffic measured in different regions of the world [111], Peer-to-Peer Networks still constitutes a significant fraction of the total Internet traffic.

Peer-to-peer (P2P) is a distributed system architecture in which all participants, the so called peers, are equally privileged users of the system. A P2P system forms an overlay network on top of existing communication networks (e.g., the Internet). All participating peers of the P2P system are the nodes of the overlay network graph, while the connections between them are the edges. It is possible to extend this definition of edges in the overlay network graph to all known peers, in contrast to all connected peers. Based on how peers connect to each other and thus build the overlay network, we can classify P2P systems into two basic categories:

**Unstructured**: The P2P system does not impose any structure on the overlay network. The peers connect to each other in an arbitrary fashion. Most often peers are chosen randomly. Content lookups are flooded to the network (e.g., Gnutella), resulting in limited scalability, or not offered at all (e.g., plain BitTorrent).

**Structured**: Peers organize themselves following certain criteria and algorithms. The resulting overlay network graphs have specific topologies and properties that usually offer better scalability and faster lookups than unstructured P2P systems (e.g., Kademlia, BitTorrent DHT).

The overlay network is mainly used for indexing content and peer discovery while the actual content is usually transferred directly between peers. Thus, the connection between the individual peers has significant impact on both the direct content transfers as well as the performance of the resulting overlay network. This has been shown in previous studies and multiple solutions to improve the peer selection have been proposed [10, 18, 41, 165, 177] which are described in detail in Chapter 5.3.1.

To construct an overlay topology unstructured P2P networks usually employ an arbitrary neighbor selection procedure [163]. This can result in a situation where a node in Frankfurt downloads a large content file from a node in Sydney, while the same information may be available at a node in Berlin. While structured P2P systems follow certain rules and algorithms, the information available to them either has to be inferred by measurements [146] or rely on publicly available information such as routing information [149]. Both options are much less precise and up-to-date compared to the information information an ISP has readily at hand. It has been shown that P2P traffic often crosses network boundaries multiple times [9, 86]. This is not necessarily optimal as most network bottlenecks in the Internet are assumed to be either in the access network or on the links between ISPs, but rarely in the backbones of the ISPs [16]. Besides, studies have shown that the desired content is often available "in the proximity" of interested users [86, 145]. This is due to content language and geographical regions of interest. P2P networks benefit from increasing their traffic locality, as shown by Bindal et. al [34] for the case of BitTorrent.

P2P systems usually implement their own routing [20] in the overlay topology. Routing on such an overlay topology is no longer done on a per-prefix basis, but rather on a query or key basis. In unstructured P2P networks, queries are disseminated, e.g., via flooding [73] or random walks, while structured P2P networks often use DHT-based routing systems to locate data [163]. Answers can either be sent directly using the underlay routing [163] or through the overlay network by retracing the query path [73]. By routing through the overlay of P2P nodes, P2P systems hope to use paths with better performance than those available via the Internet native routing [20, 152]. However, the benefits of redirecting traffic on an alternative path, e.g., one with larger available bandwidth or lower delay, are not necessarily obvious. While the performance of the P2P system may temporarily improve, the available bandwidth of the newly chosen path may deteriorate due to the traffic added to this path. The ISP has then to redirect some traffic so that other applications using this path can receive enough bandwidth. In other words, P2P systems reinvent and re-implement a routing system whose dynamics should be able to explicitly interact with the dynamics of native Internet routing [87, 156]. While a routing underlay as proposed by Nakao et al. [121] can reduce the work duplication, it cannot by itself overcome the problems created by the interaction. Consider a situation where a P2P system imposes a lot of traffic load on an ISP network. This may cause the ISP to change some routing metrics and therefore some paths (at the native routing layer) in order to improve its network utilization. This can however cause a change of routes at the application layer by the P2P system, which may again trigger a response by the ISP, and so on.

Peer-to-Peer systems have been shown to scale application capacity well during flash crowds [178]. 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. In case of video streaming, the video might simply no longer be available or the number of available peers is too low to sustain the required video bit-rate, resulting in gaps or stuttering of the video stream. Another challenge stems from the fact that in P2P systems peers can choose among all other peers to download content from but only if the have the desired content. Thus, the problem of getting content in a P2P system is actually two-fold: first the user needs to find the content and once it knows of possible peers it can download the content from, it needs to connect to some of them to get the desired content. Therefore, the overhead for locating and sometimes also transferring content in a P2P overlay network causes P2P traffic often to starve other applications like Web traffic of bandwidth [158]. This is because most P2P systems rely on application layer routing based on an overlay topology on top of the Internet, which is largely independent of the Internet routing and topology [9]. As a result P2P systems use more network resources due to traffic crossing the underlying network multiple times.

# 3

# Content Delivery Infrastructures: Architectures and Trends

The previous chapter provided us with the necessary background information to understand how content delivery in the Internet works in general: the protocols and technologies utilized by CDIs and the underlying network structures of the Internet. We now turn our attention on the different architectures and upcoming trends in the content delivery business in more detail. The special focus lies in the different server deployment strategies used by todays content delivery infrastructures.

This chapter consists of three parts: First, we discuss the challenges each party involved in the technical process of content delivery faces, namely the network operators (ISPs) and the Content Delivery Infrastructures (CDIs). Second, we give an overview of the deployment strategies of current content delivery architectures and discuss their advantages and drawbacks. Third, we describe emerging trends in CDI architectures and how they tackle the previously explained challenges.

## 3.1 Challenges in Content Delivery

Even today, decades after the launch of commercial content delivery, the challenges in content delivery are still manifold and affect everyone involved in the process of delivering content to end-users around the world.

The tremendous growth of traffic in recent years is boon and bane of network operators around the world. On the one hand, the increased demand for content, such

| Architecture<br>Benefit | Centralized | Datacenter | Distributed | ISP operated | Hybrid | Licensed | Application | Meta | Federated |
|---|---|---|---|---|---|---|---|---|---|
| End-User Location Available | - | - | - | + | o | + | - | - | ? |
| Network Information Available | - | - | - | + | o | o | - | o | ? |
| Deployment Agility | - | - | - | o | o | o | - | ? | ? |
| Network Footprint | - | o | + | ∼ | o | ∼ | + | + | ? |
| Network Integration | - | o | + | + | + | + | + | ? | ? |
| System Complexity | - | o | + | o | + | o | + | ? | + |
| Content Provider Business Relationship | + | + | + | - | + | + | + | o | ? |
| End-User Business Relationship | - | - | - | + | o | + | - | ? | ? |

- = *low* o = *medium* + = *high* ∼ = *limited* ? = *unknown*

Table 3.1: Benefits and drawbacks of classical and emerging CDI architectures.

as high definition video streaming or rich media websites, is one of the major drivers for end-user to upgrade their Internet access speeds. On the other hand, network operators find that the sheer amount and high volatility of traffic originating from content delivery poses a significant traffic engineering challenge and thus complicates the provisioning of the network [123].

The challenges content delivery systems are faced with are based on the fact that they are largely unaware of the underlying network infrastructure and its conditions. In the best case, the CDI can try to infer the topology and state of the network through measurements, but even with large scale measurements this is a difficult and error prone task, especially if accuracy is necessary. Furthermore, when it comes to short-term congestion and/or avoiding network bottlenecks, measurements are of no use. While many collaborative approaches have been proposed [10,41,138,177] to tackle this issue, we will elaborate on such solutions in Chapter 5.3.1, none of them is in operational use yet.

In the following, we describe the challenges network operators and CDIs face in more detail. In addition Table 3.1 summarizes the benefits and drawbacks of the different architectures presented in this Chapter.

### 3.1.1 Network Operators (ISPs)

ISPs face several challenges regarding the operation of their network infrastructure. With the emergence of content delivery, and especially with the distributed nature of content delivery, be it from CDIs or P2P networks, these operational challenges have increased manifold.

**Network Provisioning:** Network provisioning is an iterative process that encompasses the planning, design, deployment and operation of network infrastructure. It aims at ensuring normal day to day operations as well as meeting the needs of subscribers and operators of possible new services in the future. This process includes proper dimensioning of core routers, link capacities as well as establishing or upgrading peering links and locations with other network operators. Adequate network provisioning depends on realistic traffic demand forecasts in terms of volume and origin. However, with the emergence of CDIs and P2P networks, network provisioning has become much more complex. The sheer amount of traffic generated by such infrastructures and especially the high volatility of such traffic poses a significant challenge for any network planning.

**Volatile Content Traffic:** CDIs and P2P networks strive to optimize their own operational overhead and thus choose the most suitable server or peer based on their own criteria. As a result, traffic originating from content delivery is highly volatile, both spatial and temporal. With highly distributed CDIs and global scale P2P networks it becomes increasingly difficult for ISP to predict where traffic enters the network at what time and in which quantities diminishing the value of additional peering locations. Time-wise short-lived demand surges, called flashcrowds, and a much higher demand during peak hours, also known as diurnal traffic pattern, complicate things further as provisioning for peak demand becomes economically infeasible. Together, these effects also have a direct implication on the traffic engineering capabilities of ISPs: traffic engineering is usually based on traffic predictions from past network traffic patterns and requires some time to take effect.

**Customer Satisfaction:** Regardless of the increased difficulty with network provisioning and traffic engineering, end-users are demanding more and larger content, especially since the availability of high definition video services such as Netflix and YouTube. Coupled with the dominant form of customer subscriptions, flat rate based Internet access tariffs, the pressure on ISPs to reduce capital and operational costs, e.g., delay network upgrades or reduce management complexity, to keep prices competitive is enormous. Yet, pushing network utilization too far increases, e.g., packet loss and delay, and drastically reduces the Quality of Experience (QoE) of the end-user. This in turn paints a negative picture of the ISP in question and encourages end-users to cancel their subscription or switch to another provider with a better reputation.

### 3.1.2 Content Delivery Infrastructures (CDIs)

Economics, especially cost reduction, is a main concern today in content delivery as Internet traffic grows at a annual rate of 30% [43]. Moreover, commercial-grade applications delivered by CDIs often have requirements in terms of end-to-end delay [96]. Faster and more reliable content delivery results in higher revenues for e-commerce and streaming applications [102, 128] as well as user engagement [59]. Therefore, the network latency between the end-user and the CDI server is the key metric for optimizing the infrastructure. Although CDIs go to great lengths to further improve end-user performance, major obstacles in content delivery still exist.

**Network Bottlenecks:** Despite their efforts to discover end-to-end characteristics between servers and end-users to predict performance [96,128], CDIs have limited information about the actual network conditions. Tracking the 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 the characteristics of the network paths between the CDI servers and the end-user, the CDIs end-user assignment can lead to additional load on existing network bottlenecks, or even create new ones.

**End-User Mis-location:** DNS requests received by the CDIs authoritative DNS servers originate from the DNS resolver of the end-user, not from the end-user themselves. The assignment of end-users to servers is therefore based on the assumption that end-users are close to their DNS resolvers. Recent studies have shown that in many cases this assumption does not hold [6, 112]. As a result, the end-user is mis-located and the server assignment is not optimal. As a response to this issue, two DNS extensions have been proposed to include the end-users IP [48] or subnet information [47, 132].

**Limited Deployment Agility:** To cope with the ever increasing demand for content CDIs have deployed massively distributed infrastructures. But increasing the network footprint is becoming increasingly challenging. On the one hand the management overhead for deploying additional servers inside a network takes significant time and effort due to contract negotiations, limited space and power supply in aggregation points and intense competition, sometimes even by the ISPs. On the other hand the traffic demand is extremely volatile, especially because peak traffic is the fastest growing part, which makes provisioning difficult and finding the right location for the server even harder. A location that might look good today might be underutilized in the future, but the contracts usually run for long periods of time.

**Content Delivery Cost:** Finally, 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 [108, 143]. 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. Part of the delivery cost is also the maintenance and constant upgrading of hardware and peering capacity in many locations [128].

## 3.2 Content Delivery Landscape

To cope with the continuously growing end-user demand for content and to ensure the required quality levels in content delivery, CDIs have deployed huge distributed server infrastructures that replicate and distribute popular content in many different locations on the Internet [7, 102], posing significant deployment challenges. To complicate matters further, some of these infrastructures are entangled with the very infrastructures that provide network connectivity to end-users. But not all CDIs are built upon the same philosophy, design, and technology. For example, the required infrastructure for content delivery can be deployed and operated by an *independent* third party, often referred to as *Content Delivery Network (CDN)*, with the infrastructure deployment strategies ranging from *centralized hosting* facilities, e.g., renting space in a well connected datacenter or leasing resources in a public cloud, over multiple dedicated *datacenters* in geographically disperse locations and direct connectivity to all relevant network operators in each region, to a highly *distributed* deployment of thousands of caches deep inside many different networks. A more specialized CDI architecture is *operated by ISPs* offering a more network integrated deployment but also limits the CDI footprint to the ISPs own network.

### 3.2.1 Independent Content Delivery

Content Delivery Infrastructures operated by autonomous third parties, also known as Content Delivery Networks or CDNs, are called independent CDIs because they operate their server infrastructure and deliver content independent from the underlying network that provides the necessary connectivity. Such CDIs usually either negotiate dedicated peering agreements with network operators or pay them for connectivity just as any other customer do, e.g., end-users or corporate networks. Thus, the CDI is not overly concerned with the load it imposes on the network and considers network connectivity simply a service they pay for and leave the management of the network to the operators. However, the load of the network providing connectivity has a significant influence on the end-users performance and recently both the CDIs and network operators have started to look more and more towards collaborative approaches to further optimize content delivery, see Chapter 5.3.1.

Independent CDIs have a strong customer base of content producers and are responsible for delivering the content of their customers to end-users around the world. Based on traffic volume as well as hosted content, CDIs are today by and large the biggest

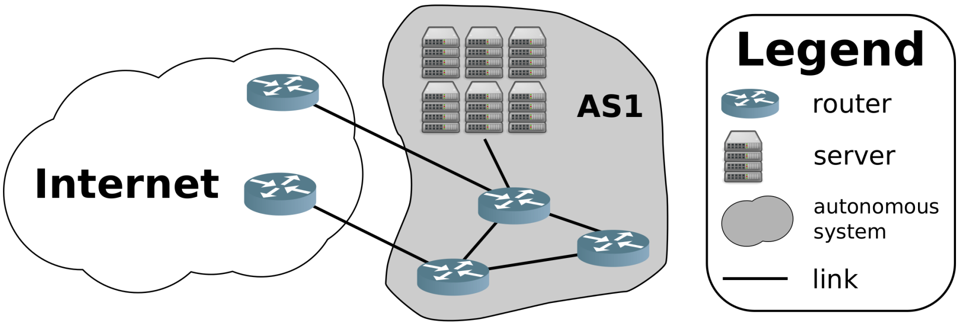


Figure 3.1: Centralized Hosting

players on the Internet, spearheading any recent traffic study and expected to grow in the future. But the content delivery market has become highly competitive with many new entrants like network operators or companies offering cloud computing. In addition, dwindling profit margins in storage and processing [22] further increase the economic pressure. To remain competitive, independent CDIs strive to increase their network footprint, optimize the performance for end-users and, probably most important, try to reduce the content delivery cost itself.

The general architecture of CDIs as described in Chapter 2.4 consists of three main components: (1) the deployment of a server infrastructure, (2) a strategy for content replication and (3) a mechanism to direct users to servers. The remainder of this section focuses on the benefits and limitations of current deployments utilized by independent CDIs [102]: centralized hosting, datacenter based, and distributed infrastructures.

**Centralized Hosting**: Centralized hosting is the most traditional deployment strategy for servers and it utilizes a single or a small number of geographical locations, e.g., co-located servers in a datacenter or rented resources from a cloud provider[1], to host and distribute content. This approach is usually used by small sites catering a localized audience, One-Click Hosters, and applications running in the public cloud.

Centralized hosting takes advantage of (a) the economies of scale that a single location offers [22], (b) the flexibility that multihoming offers [74], and (c) the connectivity opportunities that IXPs offer [5]. Using multiple geographical disperse locations provides improved performance, due to being closer to different sets of end-users, higher reliability, through redundancy, and offers scalability, by additional resources but at the same time multiplies the management overhead. Yet, for many commercial-grade applications with strict service requirements the performance and reliability falls short of expectations as the end-user experience depends on the absence of “middle mile” bottlenecks of the Internet. At the same time the

[1] Cloud providers usually operate multiple, geographical disperse datacenters that a customer can manually select when requesting new resources.

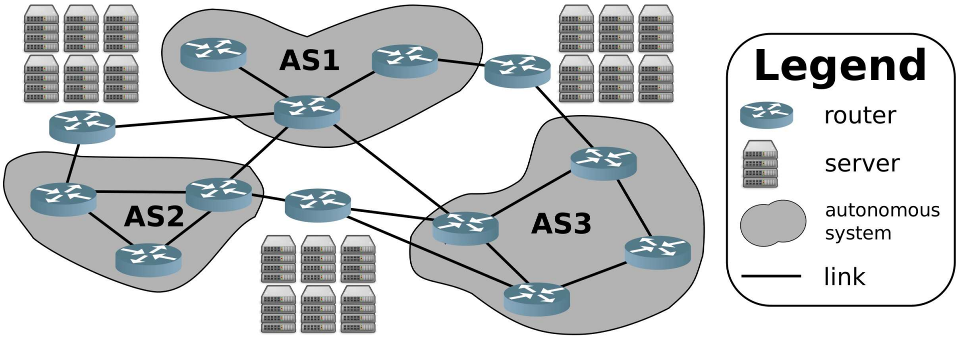


Figure 3.2: Datacenter Based

overall scalability of this deployment strategy is limited as the total capacity of a single location is limited by the existing physical space to place servers, the provided electricity and available connectivity in terms of access bandwidth to the Internet.

Another major disadvantage of centralizes hosting is the potential single point of failure, such as disrupted service due to natural desasters, distributed denial of service (DDoS) attacks on parts of the infrastructure affecting the whole deployment or limited to no connectivity in case of cut fiber-optics [102]. In addition, traffic levels fluctuate tremendously, especially during peak-hour [43], and the need to provision for peak traffic can result in underutilized infrastructure most of the time. Moreover, predicting future traffic demands accurately is rather difficult and challenging. Often, a centralized hosting architecture therefore does neither offer sufficient agility to handle unexpected demand surges nor the flexibility to scale the infrastructure for global scale operations. Moreover, it limits the CDIs ability to ensure low latency to end-users located in different networks around the world [105].

**Datacenter Based**: The datacenter based content delivery architecture can be seen as the natural evolution of the centralized hosting architecture that, simply speaking, merely multiplies the number of locations to deliver content from. The continuous demand for increased capacity and improved performance for end-users of applications and websites on a global scale has driven CDIs to increase their network footprint and delivery capacities in different regions of the world. By switching to a content delivery architecture that comprises of many large and well connected datacenters in highly populated regions in the world enables CDIs to compensate many shortcomings of centralized hosting infrastructures. The availability of multiple redundant, geographical disperse datacenters connected to major Internet backbones and the most important local networks offers reduced latency towards end-users in the region and increases the total deliver capacity of the CDI while further leveraging the economies of scale to reduce the cost for content delivery. Thus, the datacenter based content delivery architecture allows CDIs to further scale up their operations and improves the delivery of content manifold while at the same time reduces the total cost for the content delivery infrastructure.

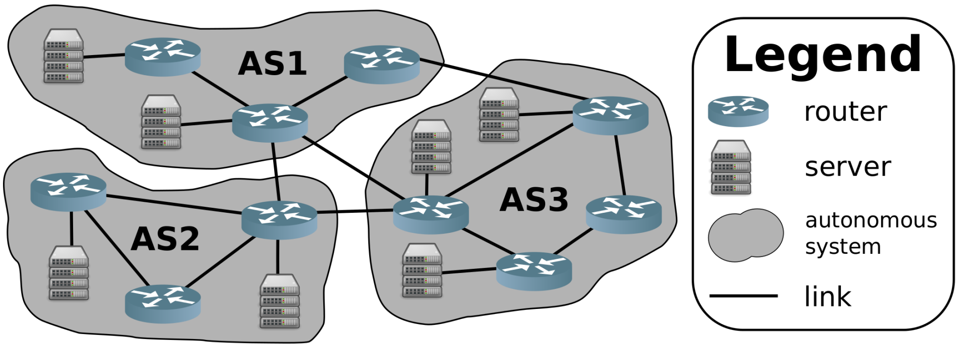


Figure 3.3: Highly Distributed

However, even with multiple well connected datacenters in each region of the world, the potential performance improvements are still limited because the CDI servers are still too far away from most of the end-users: due to the long tail distribution of end-users over all the networks that make up the Internet, the requested content for more than 50% of all users needs to traverse many “middle mile” networks, even when the CDI connects to all major Tier-1 backbones [102]. While the availability of multiple redundant datacenters offers the possibility to avoid network bottlenecks due to increased path diversity, redirecting end-users to another datacenters usually incurs major performance degradation. Also, the end-user to server assignment becomes much more important in such an architecture as selecting the correct (meaning closest) datacenter is the most crucial factor for the end-user experienced performance. Recall that assigning end-users to servers is by no means a trivial task (Chapter 2.4.3 discusses the available mechanisms including their benefits and drawbacks) and that significant improvements are possible through collaborative approaches between CDIs and network operators, see Chapter 5.3.1. Altogether, the ability to compensate sudden surges in demand is much better compared to centralized hosting but still somewhat limited and the missing agility to react in large shifts of end-user demand without sacrificing performance are the biggest drawbacks of such an architecture.

Nonetheless this type of architecture is highly popular. CDNs such as Limelight, EdgeCast, and BitGravity use it as well as many recent cloud computing deployments, such as Amazon CloudFront and Microsoft Azure.

**Distributed Infrastructures**: The third approach to scale up content delivery is using a highly distributed infrastructure: instead of deploying many servers in a few well connected locations, this architecture deploys a relatively small number of servers, usually called clusters, in many networks around the world. This approach scales the CDI vertically by deploying the infrastructure in thousands of networks rather than dozens as in the case of a datacenter based design. The smaller size and power requirements of clusters allow the CDI to push their servers deep inside the networks, usually into aggregation points, often referred to as “Point of Presence (PoP)”.

Those PoPs are located close to the end-users and still offering enough aggregated demand to highly benefit from caching the content on the CDI servers. Delivering the content directly from eye-ball ISPs to the end-users bypasses the “middle-mile” network bottlenecks and it avoids most of the peering, routing, and distance problems. At the same time reduces the total number of Internet infrastructure components it depends on for success. The highly distributed infrastructure also offers more alternative locations in case the closest server is fully utilized, e.g., during a flashcrowd, without increasing the pathlength as much as using an alternative datacenter does. This type of deployment also handles shifting demands much better, as more than one location can be used for loadbalancing. The same reason allows it to handle flashcrowds much more gracefully. Alltogether, this architecture offers very good scalability, low delays and a large overall capacity for content distribution.

However, to ensure impeccable operation, such a huge deployment must be designed to scale efficiently not only from a deployment but also from a management perspective. The deployment of individual clusters is much more complicated, as much more time is required to find appropriate locations, negotiate contracts, ship the hardware, and integrate it into the system. While this does not sound too complex, remember that in this approach we are talking about thousands of networks, each with its own policies, business units, and local customs, leading to a much higher management overhead than, e.g., in a datacenter based deployment. The deployed hardware and the stored content is also subject to local laws and regulations, which adds another dimension of complexity to content delivery based on this type of architecture. A big challenge for this deployment scenario is also the increasingly difficult task of finding additional physical space and power supply inside the aggregation points of the network. The highly distribute nature of this deployment requires this architecture to incorporate sophisticated global scheduling and load-balancing algorithms and a highly scalable and precise end-user to server mapping system. Selecting the correct server for an end-user is much more challenging as many more possible locations are available with each one possibly utilizing a different network path from the server to the end-user and thus exposing different network characteristics. The architecture also needs to include intelligent and automated fail-over and recovery methods, as e.g., replacing faulty hardware is much more time consuming and costly, due to more travel activity in combination with limited amount of possible repairs per deployment site. To enable continuous operations, the system also needs a robust and secure global scale software deployment mechanism, distributed control protocols, and in addition automated monitoring and alerting systems.

Today, Akamai is the largest independent CDI that uses this approach on a global scale. However, other large players, such as Google and Netflix, are following suit and started to deploy infrastructure deep inside ISP networks. Network operators or ISP also utilize this architecture but their deployment is limited to their own network footprint. The various implications of this are discussed in the next section. A special case of independent CDIs are free CDNs such as Coral [70], which follow a similar architectural design, but the server resources are offered by end-users or non-profit organizations.

### 3.2.2 ISP-operated CDIs

The content delivery market is steadily growing and continues to put a high burden on the network operators, either because they provide connectivity to independent CDIs or because they are a so called “eye-ball” ISP[2] hosting many end-users. At the same time, many ISPs offer today a range of services to their end-users, such as television as part of triple-play offers (Internet, Telephone and TV), Video on Demand, and cloud storage, that require large infrastructures to be delivered to the end-users. Together with the potential of generating additional revenues from content delivery has motivated a number of ISPs to build and operate their own Content Delivery Infrastructures. For example, large ISPs such as AT&T and Verizon have built their own content delivery infrastructures following the same architectural principles as independent CDIs, see previous section. The full administrative control over the network and its aggregation points is a huge potential advantage for the ISP, as they have the ability to select the best physical locations for server deployments, e.g., any PoP or other auxiliary support facility, and if necessary can even physically create new hosting facilities in a location that is predestined for hosting content delivery infrastructure.

The main difference to independent CDIs is that connectivity is provided solely through the ISP itself, including the peering and transit agreements with other network operators, and thus the deployment options are rather limited. Due to being restricted to a single network, the footprint of an ISP-operated CDI is limited to the network footprint of the ISP, thus such deployments are neither highly distributed nor globally operating solutions. While there are network operators with global network footprint, they usually do not have a global end-user customer base and thus would better fit into the independent CDI classification. To overcome this issue of a limited CDI footprint, the *Content Delivery Network Interconnection (CDNI)* working group under the umbrella of the IETF [124] is discussing how to interconnect these CDIs to boost their efficiency and coverage, for a more detailed introduction of CDNI, see Chapter 3.3.5. Another difference to CDIs operated by independent third parties is that ISPs traditionally do not have a large customer base among Content Producers and that offering a service for content delivery is traditionally not their core area of expertise.

[2] An eye-ball ISP offers Internet access to end-customers via a range of different technologies e.g., xDSL, Cable or dial-up.

## 3.3 Emerging Trends in CDI Architectures

Economics, especially cost reduction, is the key driving force behind emerging 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 [22] 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 sudden changes in demand. CDIs are therefore looking for new ways to expand or shrink their capacity, on demand, and especially at low cost.

### 3.3.1 Hybrid Content Delivery

In a hybrid CDI, end users download client software that assists with content delivery. 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 can take over the content delivery job entirely. Akamai already offers NetSession [3], a hybrid CDI solution for delivering very large files such as software updates at lower cost to its customers. Xunlei [56], 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 [80] shows 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. At the same time, hybrid content delivery offers greatly reduced operational cost for the CDI and promises additional savings due to less required infrastructure.

### 3.3.2 Licensed CDIs

Licensed CDIs have been proposed to leverage the benefits of combining the large content-provider customer base of an independent CDI with the large amount of end-user in an ISP [164]. In an licensed CDI, a strategic partnership between a CDI and an ISP, the ISP owns and operates the hardware while the CDI provides the content delivery software and integrates them into its delivery infrastructure. The revenue derived from content producers is then shared between the two parties. This allows an independent CDI to expand its footprint deep inside an ISP network without investing in hardware, incurring lower capital expenses and operational costs. The ISP benefits from acquiring the software for a reliable and scalable content delivery infrastructure without expensive and time consuming development. Yet, more importantly, a licensed CDI solution alleviates the need to directly negotiate with content producers, which might be challenging given an ISPs limited footprint.

### 3.3.3 Application-based CDIs

Recently, some popular applications started to generate so much traffic that the content producers are able to amortize content delivery costs better by rolling out their own, application specific CDI. Such CDIs can also be optimized to fit the needs of the delivered application much better than any other general purpose CDI, e.g., in the case of video streaming. Google with its YouTube service is a prime example for such an application based CDI. On the one hand, it has deployed thousands of servers in tens of data centers and interconnects them to a large number high speed backbone networks via *Internet eXchange Points (IXPs)* and also via private peerings directly to large ISPs. On the other hand, Google has also launched the Google Global Cache (GGC) [75], 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 and also allows an ISP to advertise (via BGP) the IP subnets of end-users that each GGC server should serve. Another example is Netflix, a high-definition video on demand streaming service which is responsible for a significant fraction of the Internet traffic in North America during peak hour. While still largely utilizing multiple third party CDIs [2], Netflix recently started to roll out its own content delivery infrastructure called Open Connect Network [122]. As a matter of fact, a recent study [49] suggests that the deployment happens in medium sized and regional ISPs and thus augments the infrastructure deployment of the already used third party CDIs. Netflix, just as Google, also offers an interface where ISPs can advertise (via BGP) their preferences on which subnets are served by which Open Connect Network servers.

### 3.3.4 Meta-CDIs

Today, content producers contract with multiple CDIs to deliver their content. To optimize for cost and performance [108], meta-CDIs act as brokers to help with CDI selection. These brokers collect performance metrics from a large number of end-users to determine the best CDI for individual end-users. To this end, the brokers place small files on the different CDIs and embed request for them in popular websites, e.g., via JavaScript. When end-users visit these sites, they report back performance statistics for the different CDIs[3]. This allows the broker to recommend the best performing CDI for a request from a specific network region. Meta-CDIs can also take the cost of delivery or other important metrics into consideration. Cedexis is one of these brokers for web browsing. Another broker, Conviva [59], optimizes CDI selection for video streaming. Such brokers may select another CDI to improve the end-users performance in case a CDI does not select the optimal server (which a recent study [137] has shown sometimes occurs).

### 3.3.5 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 virtual or 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 [170]. The IETF CDNI working group [124] works on CDI federation.

## 3.4 Summary

This chapter investigates the current design and operating space of todays content delivery landscape and upcoming trends in content delivery architectures. We first discuss the challenges ISPs and CDIs face in todays content delivery. Next, we give a detailed description of the classic content delivery infrastructures, namely independent and ISP operated CDIs. We then summarize the different emerging trends in content delivery that aim to solve the discussed issues.

We find that the content delivery landscape is in a constant flux to further improve the content delivery performance, increase their network footprint and at the same try time reduce the capital investment and operational costs for their content delivery infrastructure. However, not all present challenges are about to be solved by future CDI architectures. The lack of agility in server deployment as well as limited knowledge about the state of the underlying network still offers a significant potential for improvements in content delivery. Therefore, we believe that CDI-ISP collaboration will play a significant role in the future content delivery ecosystem. In the next Chapter we will look at the current use of content delivery infrastructures in an European Tier-1 ISP to further motivate the need for collaboration in the deployment and operation of server infrastructures in content delivery.

[3] Usually a single end-user does not download all the files from all the CDIs each time he or she visits the site, but rather one random file from one of the CDIs.

# 4 CDI Measurement Study

To effectively tackle the challenges for CDIs and ISPs discussed in the previous Chapter and to understand the potential benefits of CDI-ISP collaboration, it is crucial to understand the use of CDIs "in the wild". The lack of knowledge about the underlying network limits the efficiency of the CDIs user mapping and makes infrastructure deployment more complex. However, this brings up the question of how much benefit such a CDI-ISP collaboration can potentially offer? In this chapter, we answer this question by analyzing anonymized packet level traces from a large European Tier-1 ISP as well as conducting an active measurement study to quantify the potential performance benefits of CDI-ISP collaboration for end-users.

To asses the potential benefits, we first identify the most popular services inside the ISP. We continue to analyze the traces towards identifying CDI infrastructures and their behavior as seen by an ISP. Our analysis focuses on the user to server mapping and operational behavior of CDIs. We then investigate the server location diversity of CDIs and based on these observations, we develop classification methods to infer content delivery infrastructures. We continue our study by performing a first potential analysis of CDI-ISP collaboration when basic ISP knowledge is available. Next, we shortly comment on the translation of CDI server diversity to network path diversity inside the ISP. To quantify the possible performance improvements for end-users and to highlight the potential benefits of CDI-ISP collaboration, we conduct an active measurement study of the two most popular CDIs. We leave the analysis and discussion of collaborative server infrastructure deployment for Chapter 7 as we first need to introduce the necessary enablers in Chapter 5 and present the architecture of the required system in Chapter 6.

Table 4.1: Summaries of anonymized traces.

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

## 4.1 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 [134] with dynamic protocol detection (DPD) [61].

We use an anonymized 24 h packet trace collected in March 2010 (MAR10) for detailed analysis of the protocol behavior. For studying longer term trends, we used Bro's online analysis capabilities to collect an anonymized protocol specific trace summary (HTTP-14d) spanning 2 weeks. Additionally, we collected an anonymized 5 day DNS trace (DNS-5d) in February 2010 to achieve a better understanding of how hostnames are resolved by different sites. Due to the amount of traffic at our vantage point and the resource intensive analysis, we gathered the online trace summaries one at a time. Table 4.1 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.

### 4.1.1 Popular Services

With regards to the application mix, see Table 4.1, Maier et al. [111] 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 [98, 111, 151, 155]. Figure 4.1 [154] 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.

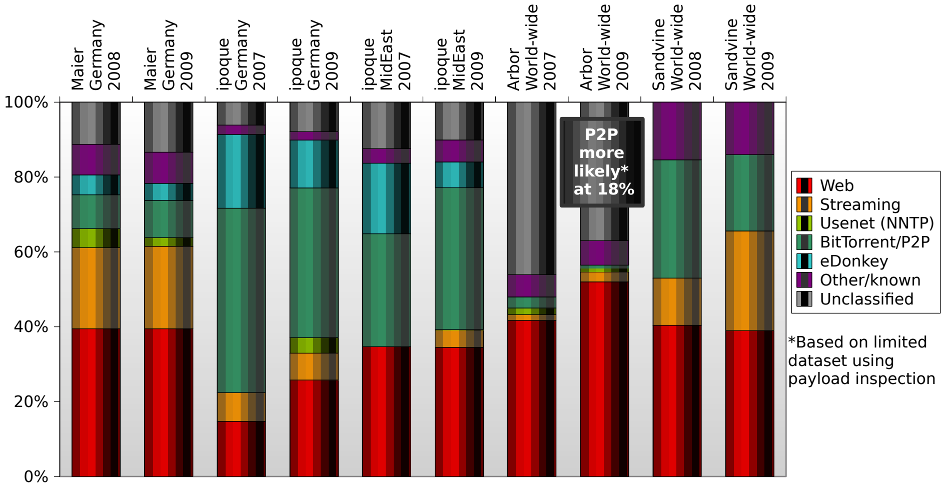


Figure 4.1: Barplot [154] of Internet Application Mix (unified categories) across years and regions from multiple sources [98, 111, 151, 155].

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 [111]. 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 are 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, 111] and in MAR10, more than 50 % of the trace's total traffic is captured by these hosts.

## 4.2 Server Diversity and DNS Load Balancing

To better understand how HTTP requests are handled and assigned to servers, we use DNS-5d to analyze the 20 most heavily queried DNS names to identify typical usage patterns. We consider only the most heavily used resolver. Figure 4.2 shows

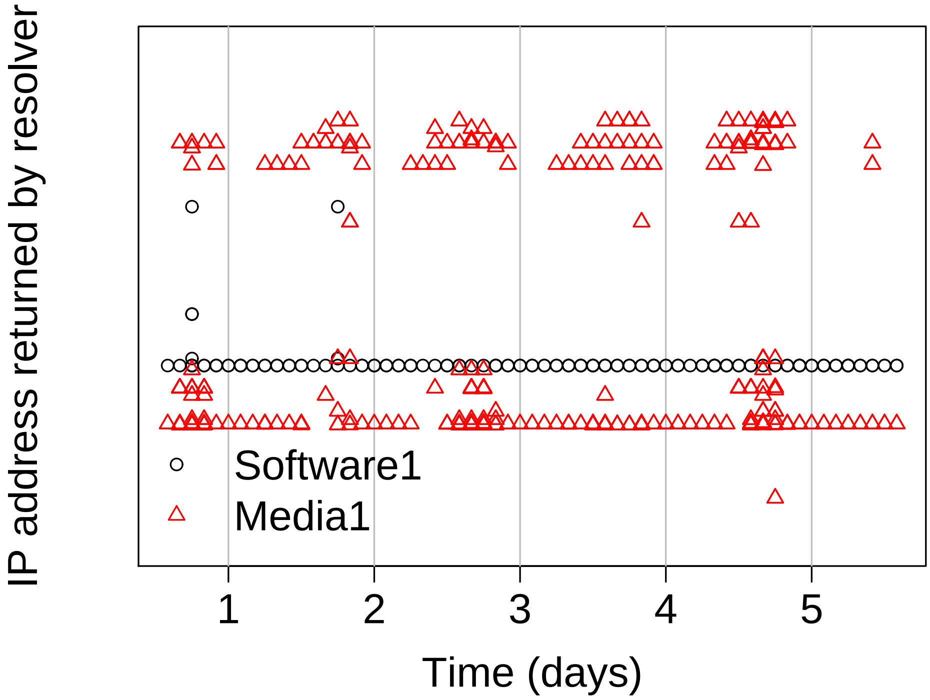


Figure 4.2: DNS replies for two sites utilizing CDIs (2h bins).

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

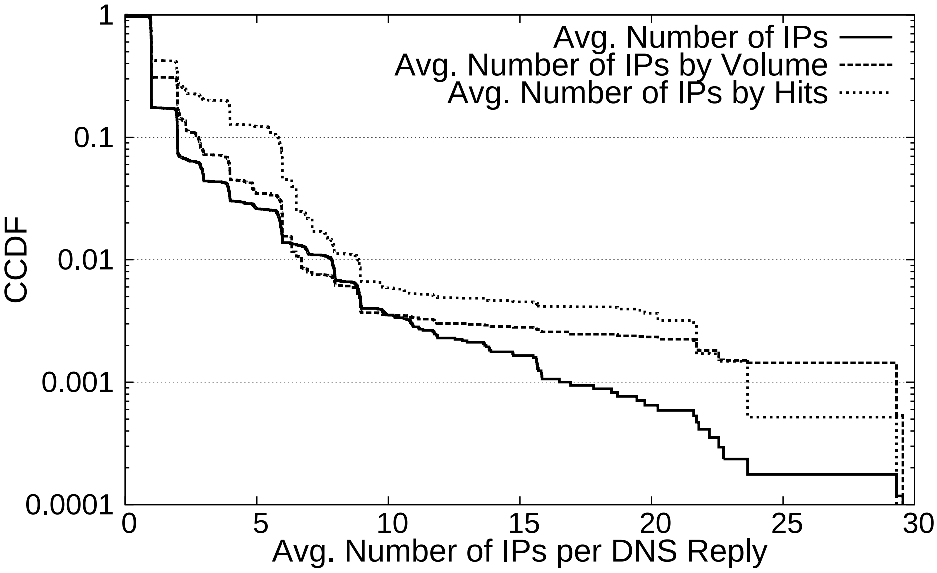


Figure 4.3: CCDF of mean # of IPs in replies from the ISPs DNS resolver.

## 4.3 Server Location Diversity

Our analysis of hostnames and their assignment to servers in Chapter 4.2 shows that content can be served by multiple servers in different locations. In fact, many domains use the service of a *Content Delivery 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 [166]. Recall, 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 are Content Distribution Networks (CDNs), such as Akamai [81,102,166], and Content Delivery Platforms (CDPs), such as Google [96] and their YouTube service [38].

To better understand the DNS resolution process for hostnames hosted on CDIs infrastructure, 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 acting as a cache. Lastly, the authoritative DNS server, henceforth referred as `DNS server`, which is usually run by the CDI, replies to the DNS resolver. The DNS server can choose to return one or more server IP addresses based on the domain name in the request and the IP address of the requesting DNS resolver. For example, it may use a geolocation database [160] to localize the region of the DNS resolver, utilize BGP data to identify the ISP, create a topology map derived via traceroutes, or any combination of these and other topological and ge-

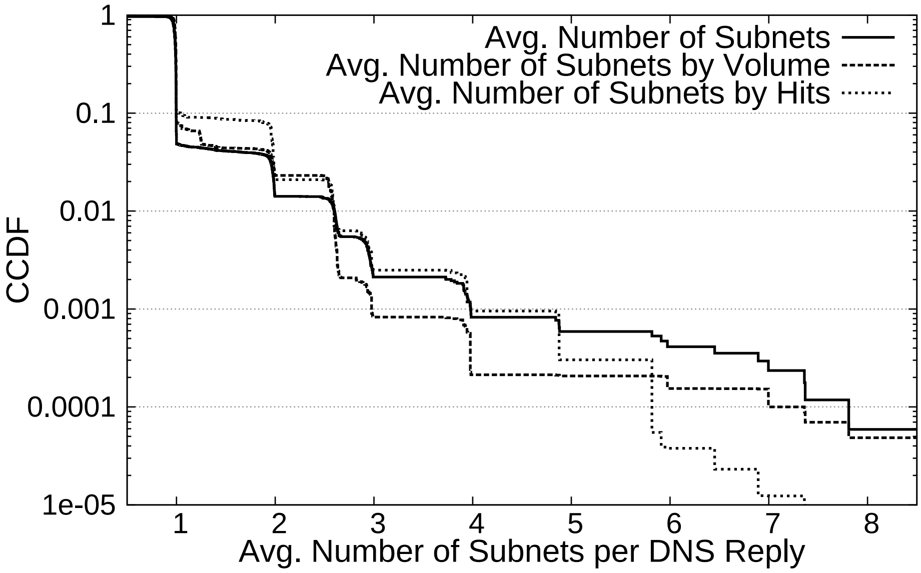


Figure 4.4: CCDF of mean # of subnets in replies from the ISPs DNS resolver.

ographic localization techniques. A DNS server has, in principle, two methods for load balancing across multiple servers:

**MultQuery:** multiple IP addresses within a single DNS response
**CrossQuery:** different IP addresses for repeated queries of the same domain

In our active DNS measurements, we find that often a mixture of MultQuery and CrossQuery is being used in practice. Furthermore, we use 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 achieve this by aggregating the returned IP addresses into subnets based on BGP information obtained from within the ISP. This allows for detailed information about the different locations within the ISP, while giving an aggregated view of subnets reachable via peering links.

Another issue stems from the fact that the IP address returned by the CDI usually depends on the IP address of the ISP DNS resolver [6, 133, 166]. Due to this, we use the DNS resolver of the ISP of our vantage point as well as external DNS resolvers (see Chapter 4.3.1). The former reflects the experience of most of the clients at our vantage point[1]. The latter lets us discover additional diversity as well as understand the preference of the CDI for this specific ISP.

**Prevalence of MultQuery:** We start our analysis by checking the prevalence of the first form of DNS based load balancing, MultQuery. Figure 4.3 and 4.4 show CCDF

[1] Using our traces we verify that more than 95 % of the clients use the ISP's DNS resolver.

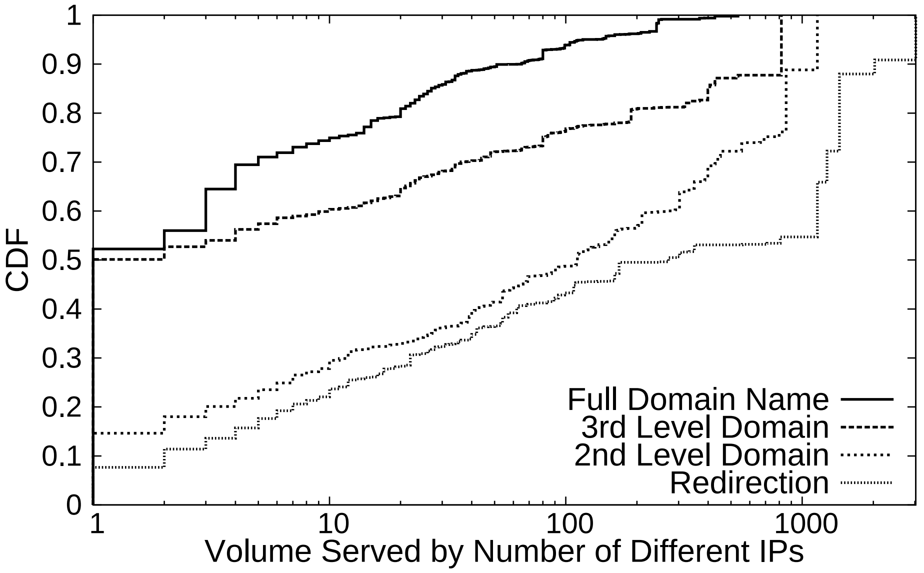


Figure 4.5: CDF of # of IPs for the ISP DNS resolver normalized by traffic.

plots of the average number of IP addresses and subnets respectively 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 $90^{th}$ percentile is 2. We note that even when an answer yields multiple IP addresses, the majority of them are from the same subnet. Therefore, the diversity decreases even further if we aggregate to subnets. From a network perspective, this implies that there is not much choice, neither for the ISP nor for the user, regarding where to download the content from. Both are limited to the information provided by the DNS server. However, when we normalize the hosts by their respective popularity, we see a significant improvement. More than 29% of the volume and 19% of requests have a choice among at least 2 IP addresses.

**Prevalence of CrossQuery:** Next, we check how prevalent CrossQuery, the second form of DNS based load balancing is. Since CrossQuery returns different IP addresses for repeated queries, its potential contribution to server diversity can only be studied by aggregating across time. The lines labeled `Full Domain Name` in Figures 4.5 and 4.6 capture this case.

We find that more than 50 % of the volume or requests can be served by more than one IP address. similarly, there is choice between at least two subnets over 40 % of the time across both metrics, see Figure 4.6. This indicates that most CDIs serve content from multiple locations.

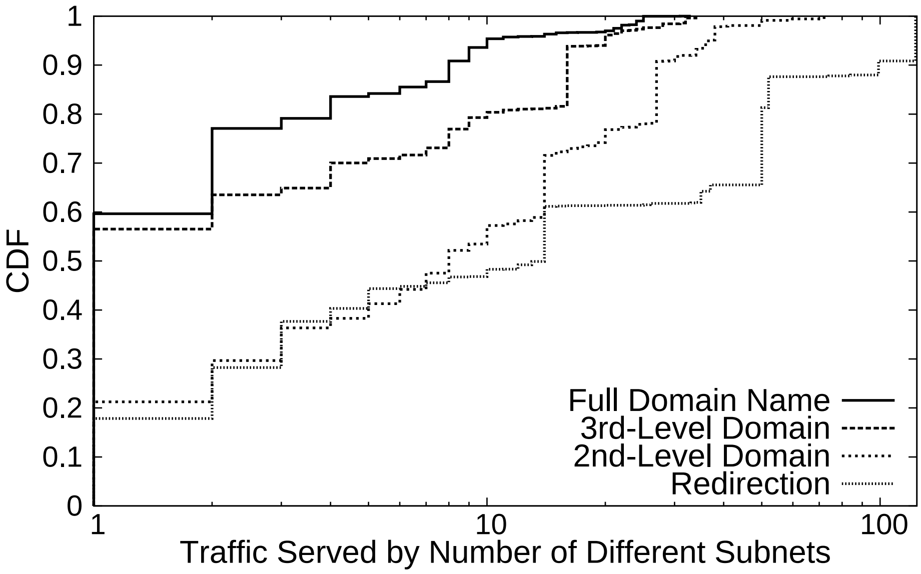


Figure 4.6: CDF of # of subnets for ISP DNS resolver normalized by traffic.

**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 4.5 and 4.6 at the respective data series called `3rd Level Domain` and `2nd Level Domain`. For example, we might accumulate the IP addresses from DNS replies for `dl1.example.org` and `dl2.example.org` for the statistics on the 2nd level domain, but not the third level domain.

This is a feasible approach, since many hosts respond to all requests that belong to a subset of the subnets returned when accumulating by the second-level domain of DNS resolver answer, including recursive requests and redirections. We verify this behavior with active measurements, see Chapter 4.5. We find that at least two major CDIs, a streaming provider and a One-Click Hoster, serve requested content from servers that match in their second level domain.

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 volume as well as by requests. Studying our traces in more detail, we find that this is due to the substantial traffic volume and number of requests that are served by CDIs, some of which are highly distributed within ISPs or located in multihomed datacenters or peer-exchange points.

**Infrastructure Redirection Aggregation:** Taking a closer look at the DNS replies, see Chapter 2.2, we find that some CDIs use CNAME records to map queried hostname to an A record. These A records show the same pattern as the hostnames in

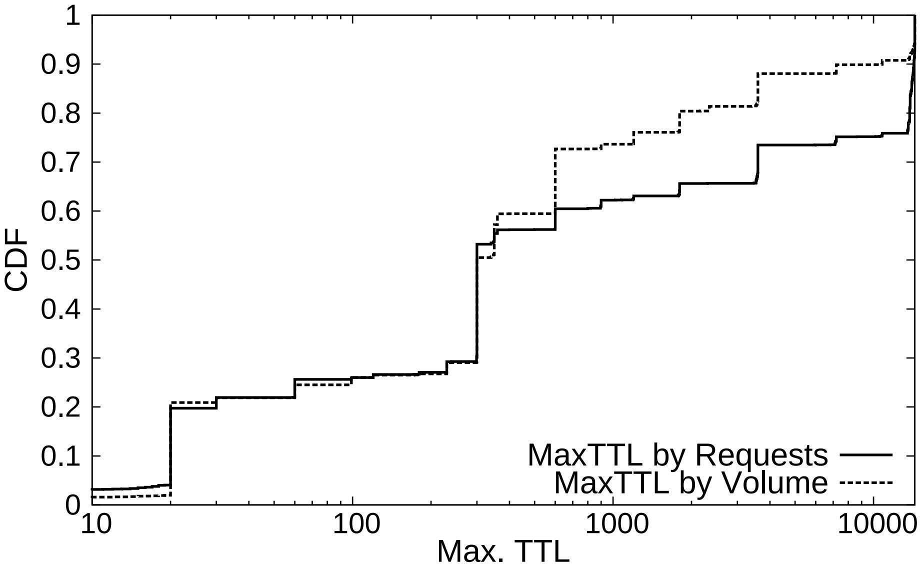


Figure 4.7: CDF of DNS TTL value by traffic volume and by number of requests.

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 [169]. 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 4.5 and 4.6.

Turning our attention to the implications of the proposed aggregation schemes, we notice the available diversity increases tremendously. More than 70% of the bytes and 50% of the hits are served by more than 20 servers. With regards to subnets, the diversity decreases slightly. Nevertheless, more than 5 subnets are available for 55% of the bytes and 45% of the hits. If we consider aggregation periods in the order of tens of minutes, the numbers do not decrease by much. The reason that most of the diversity is observable even over these short aggregation time periods, is that the typical TTL, see Figure 4.7, is rather short with a mean of 2,100 seconds and an median of 300 seconds normalized by volume. When weighted by requests, the mean is 4,100 seconds and the median is 300 seconds.

Table 4.2: Traffic localization within the network by different DNS resolvers normalized by number of requests and traffic volume together with the potentially available fraction of localized traffic.

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

### 4.3.1 Alternative DNS Resolvers

So far we have only considered the effect of content diversity when the ISP DNS resolver is used. To understand how much the DNS load balancing deployed by a CDI is biased by the queried DNS resolver, we repeat the experiment from Chapter 4.2 using two other DNS resolvers. In particular, we pick the next most popular DNS resolvers found in our traces: GoogleDNS and OpenDNS[2].

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 server 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 AS the DNS resolver belongs to, the server selection by the CDI cannot optimize for the location of the DNS client.

### 4.3.2 Impact on Traffic Localization

Analyzing the three active DNS measurements from the ISP, OpenDNS, as well as Google DNS resolver, we find that a significant part of the requests that can in principle be 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 can 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 Chapter 4.3 and account the volume (or requests) evenly to each of the IP addresses. Next, we classify the IP addresses in two groups - inside and outside of the ISP network. Table 4.2 summarizes the results of this aggregation regarding the traffic and hits that were kept inside the ISP's network in the columns labeled `observed`.

[2] Both are third-party resolvers with a global footprint and utilize IP anycast.

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

In contrast, clients that use the ISP's DNS resolver fare better with regards to AS distance: 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 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. 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 request [48].

However, one can ask if the CDI utilizes the full potential of traffic localization. For this, we check the potential of traffic localization, by changing the volume (or hit) distribution from even to greedy. Thus, as soon as we observe at least one IP address inside the ISP's network, we count all traffic for the entire aggregation to be internal. Table 4.2 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 can not only in itself result in a substantial reduction of costs for the ISP, but it also points out the benefits of CDI-ISP collaboration. While the increase is noticeable for OpenDNS, it is nowhere near that of the ISP's DNS resolver. The potential benefit when relying on GoogleDNS is rather small. A deeper study on our results unveils that content served by highly distributed and redundant infrastructure can be localized the most.

## 4.4 From Server Diversity to Path Diversity

Next, we ask the question whether the substantial diversity of server locations actually translates to path diversity. For this purpose, we generate a routing topology of the ISP by using data from an IS-IS and a BGP listener. However, due to the asymmetry of routing, we have to explore both directions separately. Following the argumentation from Chapter 4.3 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 all 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 use the routing data to determine the paths of all content that is potentially available within the ISP's AS[3]. 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.

## 4.5 Active Measurements

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

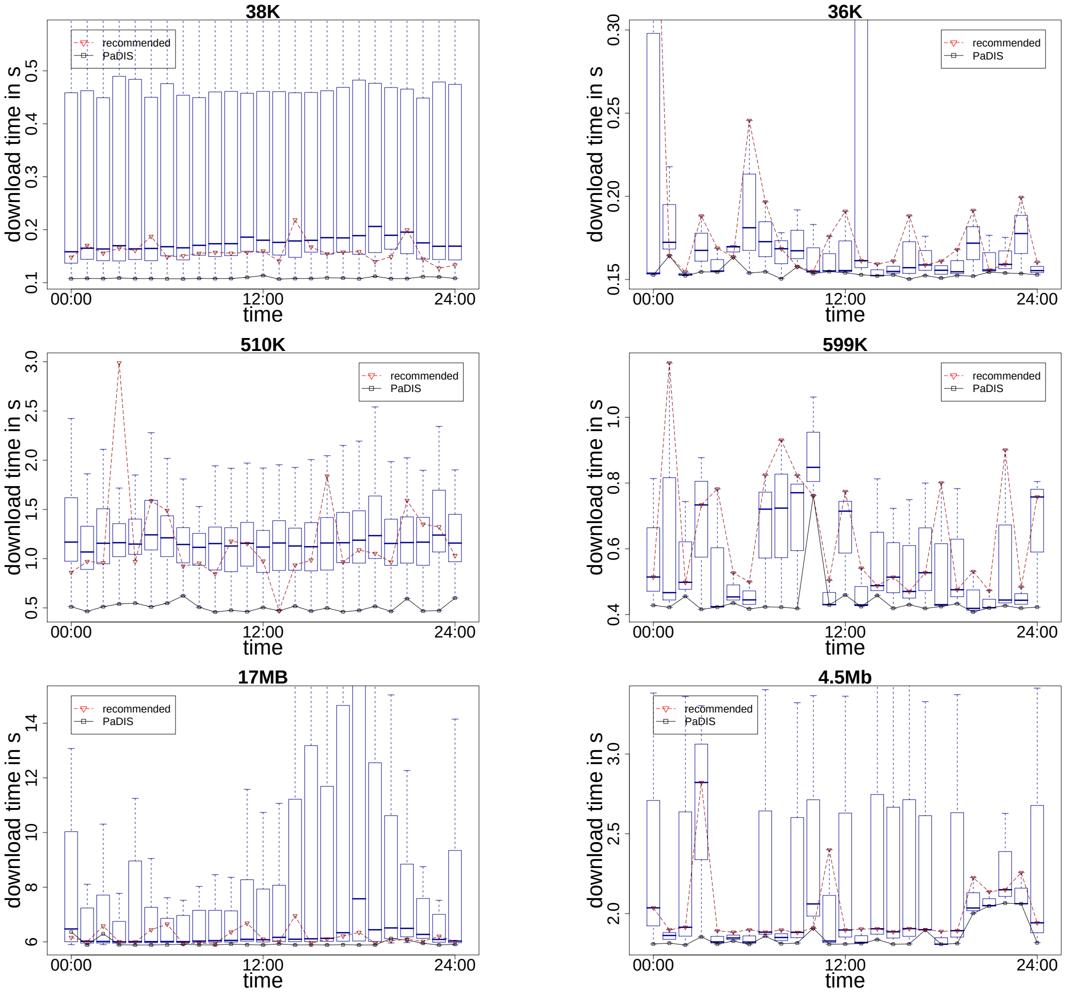
The ten vantage points are deployed within residential locations with DSL connectivity to the ISP. The downstream bandwidth ranges from 1 Mbps to 25 Mbps while the upstream ranges from 0.128 Mbps to 5 Mbps. The measurements started on 1st of May 2010 and lasted for 14 days. Each client accesses the selected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

### 4.5.1 Content Delivery Infrastructures

Using the data sets from the residential ISP, see Chapter 4.1, we identify the two most popular CDIs, referred to as CDI1 and CDI2. These are responsible for roughly 20 % of all HTTP traffic. Using the methodology discussed in Chapter 4.3, we identify more than 3,500 unique IP addresses that are caches for CDI1 and more than 700 unique IP addresses for CDI2. Both of these CDIs have more than 300 of their cache IP addresses within the ISP.

After augmenting each identified CDN IP address with its network path information, see Chapter 4.4, we find that the server diversity translates not only into subnet diversity, but also path diversity. Since recent studies of CDI behavior have shown

[3] Augmenting the routing topology with flow information may allow us to extend this analysis to all content, in contrast to the content within the ISP's AS.



(a) Object download times for CDI1.

(b) Object download times for CDI2.

Figure 4.8: File download times for CDI caches across time.

that objects are accessible from an arbitrary server [81,169], we can bypass the CDIs server selection. Thus, we request the URL directly from each of the identified CDI server IP addresses regardless of their location. We verify this for all servers of CDI1. However, CDI2 is more restrictive. Our measurements show that CDI2 servers only reply to requests from the same region. In our case, we observe that European caches do serve the content to our European clients. However, when requested from North American servers, the request was denied.

Since the download performance of Web pages depend on the size of the object, we select objects of different but comparable file sizes for both CDIs ranging from 36 KB to 17 MB, see Table B.1 in the appendix. To be able to repeat the measurements multiple times during a small time period while not overwhelming the client DSL lines, we subsample the number of servers of both CDIs. To preserve path diversity, we randomly select one server from each subnet. This reduces the number of caches to 124 for CDI1. For CDI2 we find five subnets[4] containing servers, yet only two answer our queries, as we have already explained. In addition, we download each object once per measurement by normally resolving its respective hostname, thus following the CDIs server recommendation. In this case we exclude the DNS resolution time from our measurements.

Figures 4.8a and 4.8b show boxplots of object download times during a typical day (May $12^{th}$) for one specific client, with the objects being comparable in size for both CDIs. Comparing the results from other clients and other objects, we see similar results throughout the experiment. We use box plots because they are a good exploratory tool allowing the visual inspection of typical value ranges, spread, skewness, as well as outliers. Each box analyzes the results of downloading the selected file at one point in time from one server in each of the subnets, e.g., for CDI1 each box consists of 124 data-points. The box itself stretches from the 25th to the 75th percentile. The line within the box corresponds to the 50th percentile (the median). The whiskers represent the lowest and highest datum still within 1.5 times the interquartile range of the lower and upper quartile respectively. The dashed lines with triangles corresponds to the object download time for the recommended server by the CDI. The solid line with squares corresponds to the object download time for the server that is most suited according to the ISP.

A first observation regarding Figures 4.8a and 4.8b is that the download time for the recommended servers are quite good and close to the median download time of all examined servers. Still, there is significant room for improvement especially during peak hours. Overall, there is potential to improve the download time up to a factor of four. Our active measurements also highlight typical network effects. For example, when downloading small objects, TCP is typically stuck in slow start. Thus, the round-trip time to the cache is the dominant factor for the retrieval time. When downloading medium-size objects, both bandwidth and delay matter. For large objects the performance is usually restricted by the available network bandwidth including the download bandwidth of the last-hop to the client (25Mbit/s in this experiment). For CDI1 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.

With respect to the ISP's benefits, we point out that there is potential to localize the content within the ISP and that the average path length within the AS was reduced from 3.8 to 3 when downloading content from CDI1. Due to the limited path diversity for CDI2, the internal path-length remained unchanged, even though there is the possibility to decrease the download time.

[4] Apparently, CDI2 utilizes well provisioned and well connected data centers around the world and thus relies on the redundancy within the data centers and their access to multiple ISPs.

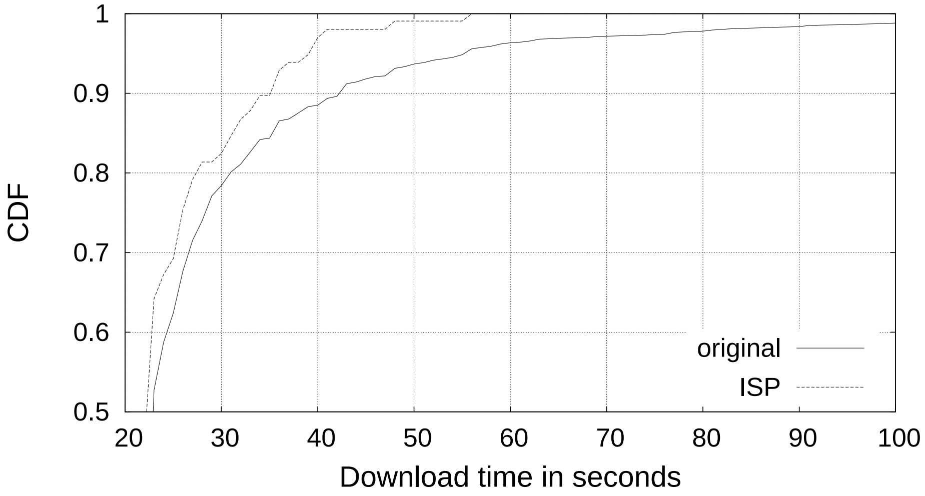


Figure 4.9: Distribution of download times of OCH1.

### 4.5.2 One-Click Hosters

One-Click Hosters (OCH) offer users the ability to share files via a server based infrastructure, typically located within one or several well-provisioned data centers. Recent studies have shown that OCHs can achieve better download time than, e.g., P2P systems such as BitTorrent [21]. Therefore, it is believed that such services may become the leading platform for file sharing and replace P2P systems. Using our data sets from the residential ISP, we identify the most popular OCH, referred to as OCH1, which is responsible for roughly 15 % of all HTTP traffic. OCH1 is located at a multi-homed data center in central Europe. To scale the number of parallel flows, OCH1, like other OCHs, limits the maximum file size to 200 MByte.

Using our traces, as well as studying the DNS naming scheme of the servers, we are able to deduce that OCH1 has twelve uplinks utilizing four different providers. The ISP we are collaborating with is among these providers. To understand how OCH1 does uplink selection, we repeatedly downloaded a 60 MByte test file during a one week period starting on the 7th of April 2010. Roughly 60 % of the requests are shown to be directed to a server that can serve the content via the direct peering with the client's ISP. From the other eleven uplinks, ten uplinks are chosen with equal probability while one is chosen with smaller probability. We also validate that it is feasible to download the file from any server of OCH1 and thus fetch the content via any of the providers. It is worth noting that there are no time-of-day or time-of-week effects at all, while the HTTP volume of OCH1 in our traces exhibits time-of-day effects. This leads us to believe that the link utilization as well as the end-user performance can be improved by jointly optimizing the server selection in a CDI-ISP collaboration.

To quantify the potential improvements for the end-user, we repeatedly download the test file from OCH1 over a period of one week. The downloads are performed every two hours for each of the 12 locations. Additionally, mapping requests are issued every 200ms to find out the dynamics in the server assignment of OCH1. Figure 4.9 shows the distribution of total download times when OCH1 assigns end-users to its servers (“original”) and compares it to the possible download times that can be achieved when utilizing network related recommendations from the ISP ("ISP"). We observe that more than 50% of the downloads do not show a significant difference. This happens mainly during non-peak hours. For 20% of the downloads, we observe a significant difference in the download times, mainly during peak hours. Together with the observation of static uplink assignment, even during peak hours, this shows that there is significant potential to improve end-user experience and enable the collaboration between ISPs and OCHs.

## 4.6 Summary

We start this chapter by asking how much benefit ISP-CDI collaboration can potentially offer. To asses the potential benefits and to understand the use of CDIs “in the wild”, we analyze anonymized packet level traces from an European Tier-1 provider. We identify the most popular services inside the ISP and find that HTTP is the dominant source of traffic, in particular since almost all video streaming is done on top of HTTP. Next, we focus on identifying CDI infrastructures. To this end, we do not only observe significant CDI server location diversity but also significant path diversity for accessing HTTP based content inside the ISP. More precisely, a first potential analysis of CDI-ISP collaboration indicates that around 50 % of the HTTP traffic can be fetched from CDI servers inside the ISP. Therefore, CDI-ISP collaboration improves the traffic localization potential more than two-fold. In addition, up to 70 % of the traffic can be transferred from alternative locations using different network paths and thus collaboration enables new mechanisms for managing traffic flows inside the network. Moreover, our active measurement study shows that utilizing ISP recommendation can improve the end-users performance for specific CDIs by up to a factor of 4 and thus highlighty the potential performance benefits of CDI-ISP collaboration.

The CDI measurement study we present in this Chapter shows the potential of CDI-ISP collaboration in content delivery. We quantify the effectiveness of collaboration in terms of possible traffic localization and improvements in end-user performance. The results strengthen our believe that a collaborative approach in content delivery enables CDIs and ISPs to jointly tackle the challenges outlined in Chapter 3. To this end, the next Chapter introduces two key enablers for solving these issues and we present our system design implementing them in Chapter 6.

# 5 System Design for CDI-ISP Collaboration

Our measurement study, see Chapter 4, shows ample opportunities for CDI-ISP collaboration to improve content delivery performance by leveraging the already high path diversity of existing server deployments. In addition, a collaborative content delivery approach lets CDIs and ISPs jointly tackle the deployment problem, regardless whether a CDI utilizes traditional or emerging solutions, see Chapter 3.

The common denominator is the lack of information about the other parties: Today's content delivery landscape is mostly unaware of information ISPs have about dynamic network conditions and end-user locations in the network. The ISPs on the other hand have no knowledge about the CDIs strategy when assigning end-users to servers or deploying new infrastructure and thus have a hard time in properly provisioning and operating their network.

However, this information influences the efficiency and performance of content delivery and while some of this information can be inferred by each party on its own [128, 137] it is a tedious and error prone task. This is especially cumbersome, as the very information is ready at hand of each respective party.

To this end, we propose in this Chapter two key enablers, namely *in-network server allocation* and *informed user-server assignment*, to facilitate the collaboration between CDIs and ISPs and to address the challenges they face without revealing sensitive operational information. We continue by presenting both enablers, outline

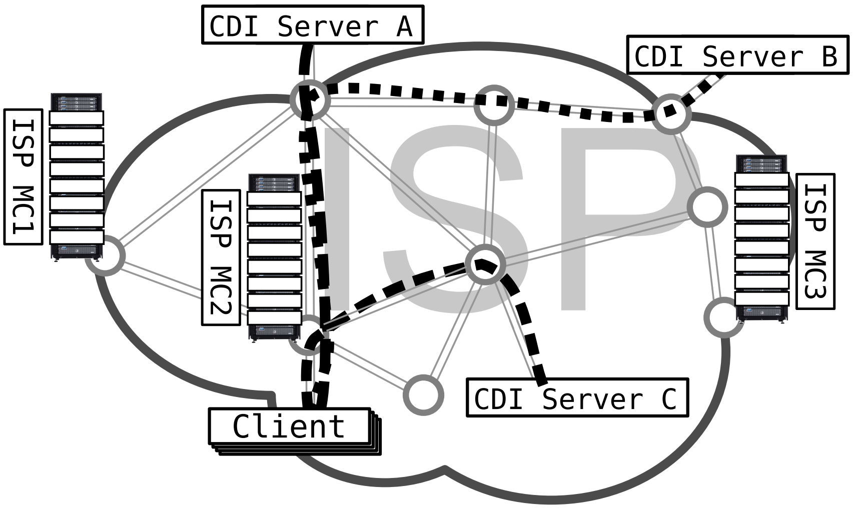


Figure 5.1: *In-network Server Allocation:* A joint in-network server allocation approach allows the CDI to dynamically expand its footprint using additional and more suitable locations (e.g., microdatacenters MC1, MC2, MC3) inside the network to cope with volatile demand.
*Informed User-Server Assignment:* Assigning a user to an appropriate CDI server among those available (A, B, C), yields better end-user performance and enables traffic engineering.

their design rationale, and propose algorithms to realize them. We present our implementation of the two key enablers, called NetPaaS, in Chapter 6, and evaluate them in Chapter 7.

## 5.1 Key Enablers

Recent trends and studies in content delivery outline a clear trend: the demand for content is rapidly growing and as a result CDIs have an increased need of server resources close to the end-users to satisfy their demand with acceptable performance. However, the challenges as described in Chapter 3 are multifold and limit the ability of both the CDI and the ISP to efficiently operate and scale up their respective infrastructure. To alleviate those challenges and to enable both the CDI and the ISP to improve current and future operation, content delivery systems have to address two fundamental problems.

The first is the *server allocation problem*, i.e., where to place the servers and content. The key enabler is *in-network server allocation*, or in short *in-network server allocation*, where the placement of servers within a network is coordinated between CDIs, ISPs, and content providers. This enabler provides an additional degree of freedom to the CDI to scale-up or shrink the footprint on demand and thus allows it to deliver content from additional locations inside the network. Major improvements in content delivery are also possible due to the fact that the servers are placed in a way that better serves the volatile user demand. The application of this enabler is two-fold. One, it helps the CDI in selecting the locations and sizes of server clusters in an ISP when it is shipping its own hardware. The second application is suitable for more agile allocation of servers in cloud-style environments, such as those mentioned in [123]. Multiple instances of virtual servers running the CDI software are installed on physical servers owned by the ISP. As before, the CDI and the ISP can jointly decide on the locations and the number of servers. A big advantage of using virtual machines is that the time scale of server allocation can be reduced to hours or even minutes depending on the requirements of the application and the availability of physical resources in the network.

The second enabler is the *end-user to server assignment problem*, i.e., how to assign users to the appropriate servers. The key enabler for addressing this problem is *informed user-server assignment* or in short *informed user-server assignment.* It allows a CDI to receive recommendations from a network operator, i.e., a server ranking based on performance criteria mutually agreed upon by the ISP and CDI. The CDI can utilize these recommendations when making its final decision regarding end-user to server assignments. Moreover, its design allows the coordination of CDIs, content providers and ISPs in near real-time, as we elaborate in Chapter 6. Any type of CDI can benefit from this enabler, including ISP-operated CDIs and P2P systems. The opportunities for this enabler are multifold as it addresses multiple challenges. The recommendation allows the ISP to take possible network bottlenecks into account and at the same time enables the ISP to influence how the traffic flows through its network thus reducing the network traffic volatility. In addition it has precise knowledge about the end-users location and the current network conditions and thus it can effectively select the best possible candidate server for each individual end-user request. As a result, the improved performance increases the customer satisfaction and simplifies future network provisioning and traffic engineering. A major advantage of this enabler is that in comparison with other CDI-ISP [58, 84] cooperation schemes no routing changes are needed which reduces the network management complexity. We provide the high-level intuition for both enablers in Figure 5.1.

Until now, both problems have been tackled in a one-sided fashion by CDIs. We believe that content delivery can be improved via accurate and up-to-date information during the server selection by the CDI. This also eliminates the need for CDIs to perform cumbersome and sometimes inaccurate measurements to infer the ever changing network conditions within the ISP. We also believe that the final decision must still be made by the CDI. In this thesis, we argue that the above enablers (a) are necessary to enable new CDI architectures that take advantage of server virtualization technology, (b) allow fruitful coordination between all involved parties, including CDIs, CPs, and ISPs (c) enable the launch of new hand highly demanding applications jointly by CDIs and ISPs, and (d) can significantly improve content delivery performance. Such performance improvements are crucial as reductions in user transaction time increase revenues by significant margins [91].

## 5.2 In-Network Server Allocation

Recent advances in virtualization offer CDIs an additional degree of freedom to scale-up or shrink the footprint on demand. This can be either done by jointly deploying and operating new servers with the ISPs or by leveraging already existing server infrastructure inside the network. In this section we formally introduce the design of in-network server allocation motivated by the recent announcement of major ISPs to support generic hardware appliances, also referred to as microdatacenters, and offer them to application, service, and content providers. Our design of in-network server allocation leverages the view of the ISP about the network and additional computation and storage resources inside the network to enable a joint optimization of server deployments and allows the CDI to dynamically scale up or reduce their infrastructure footprint inside the ISP.

### 5.2.1 The New Cloud

Applications are increasingly relying on direct interactions with end-users and are very sensitive to delay [102]. Indeed, transaction delay is critical for online businesses [91]. Network delay and loss are important contributors to the transaction delay. Today, large-scale service deployments are restricted by limited locations in the network, e.g., datacenters, peering locations, or IXPs. These locations are not necessarily ideal [105]. We point out that *selection of service location is critical and currently not flexible enough.* Services should be located close enough to, in terms of network distance, the end-users. Since end-user demands are volatile and change across time, CDIs need more agility [42]. They can improve their service quality by quickly allocating, de-allocating, and migrating resources on-demand where and when they are needed. Indeed, since delay and packet loss are among the critical metrics, the service may need to be deployed deep inside the network, as many ISPs already do for their own IPTV services. However, this option is not yet available for non-ISP service providers, e.g., Content Delivery Infrastructures.

Currently, most services and networks are run by independent entities with different and often conflicting objectives. Lack of information about the other entity leads to suboptimal performance and resource allocation for both the CDI and the ISP. For example, CDIs implement sophisticated methods to infer network conditions to improve perceived end-user experience [128], e.g., active measurements within the ISP networks. Yet, the information gleaned from these measurements is already available with far greater precision to the ISP. Therefore, ISPs continuously upgrade their infrastructures without being able to efficiently engineer the voluminous traffic flows [137] of e.g., CDIs. Today, cooperation and/or partnership between providers is limited to, e.g., peering or lately direct interconnections with Content Delivery Infrastructures. This level of cooperation is too narrow to reduce operational costs, improve end-user experience, circumvent bottlenecks, handle flash crowds, and adapt to changing network conditions and end-user demands. This has led to initial discussions on how to improve communication between the various entities, e.g., within the IETF ALTO and CDNI working groups.

### The ISPs Proposal

To overcome the above mentioned obstacles in service deployment and operation, major ISPs, including AT&T, Verizon, Deutsche Telekom, Telefonica, NTT, have proposed the use of cloud resources consisting of general purpose appliances that are co-located at network aggregation points inside the ISP. With the convergence of computing, storage, and communications, the acceptance of cloud services, and the ever increasing demand for popular services, ISPs are moving towards deploying general-purpose computing and storage infrastructures in their points of presences (PoPs). Henceforth, we refer to these as *microdatacenters*. The description of the functionality of these microdatacenters is provided in a white paper [123] that appeared in the SDN and OpenFlow World Congress in October 2012 and was signed by 13 of the largest ISPs. Microdatacenters can be also the technical solution needed to materialize recent alliances of major CDIs, such as Akamai with large ISPs in the area of content delivery [11, 14, 15]. We notice that Software Defined Networks (SDNs) is another alternative to redirect traffic or perform traffic engineering when applied within an ISP or between and ISP and a CDN in cooperation. The comparison of the two approaches, NFV and SDN, is out of the scope of this thesis and we refer the reader to the related literature on SDN, e.g., [36, 78, 83, 115, 144].

Figure 5.2 illustrates the basic idea of in-network server allocation. The ISP can offer *slices* within its microdatacenters, that can be leased by CDIs—using our proposed mechanism—based on their needs. This approach leverages recent advances in virtualization technology, and flexible billing models, such as pay-as-you-go, to provide cost-efficient and scalable service deployment, enabling unprecedented flexibility. Moreover, the diversity of available service locations within the network can be used to improve end-user experience and makes it possible to launch even more demanding applications, such as interactive ones. In-network server allocation enables CDIs to rely on a fixed infrastructure deployment for their baseline operation and then scale it up by dynamically allocating resources closer to end-users. It also lowers the burden of entrance in the service market for smaller CDIs who can rely exclusively on in-network server allocation at first.

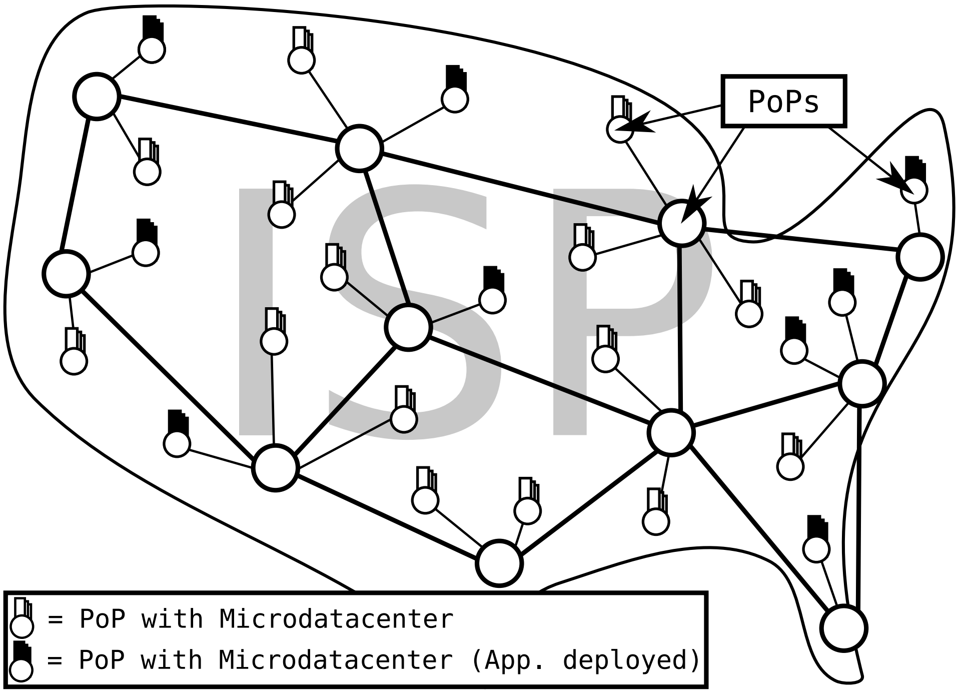

PoPs
ISP
= PoP with Microdatacenter
= PoP with Microdatacenter (App. deployed)

Figure 5.2: Microdatacenters in an ISP with In-Network Server Allocation enabled.

### Microdatacenter Specifications

The term Microdatacenter stems from the fact that every aspect of a normal datacenter is provided by a single or more industry standard server rack cabinets: network access, computation resources and storage. The main benefit of the small form factor is the ability to host Microdatacenters in locations that have limited space and power supply to accommodate infrastructure as they were not designed for this purpose, such as network aggregation points or internet exchange facilities. Microdatacenters consist of one or more racks of off-the-shelf hardware deployed in general purpose rack space at network aggregation points. State-of-the-art solutions have been proposed by the VMware/Cisco/EMC VCE consortium [171], and are also offered by other vendors, such as NetApp and Dell. These solutions are general-purpose and provide a shared infrastructure for a large range of applications. Microdatacenters typically consist of two basic components: hardware and management software.

**Hardware:** Typical microdatacenters include *storage*, *computing*, *memory*, and *network access* components. Storage consists of tens of Terabytes with an ultra-fast controller providing I/O throughput in the order of hundreds of Gbps. The storage component is connected to the Internet through multi-Gbps interfaces and to the computing component with Gigabit Ethernet switches. Typically, a rack includes up to 40 physical multi-core blade servers as well as two routers and two switches in mesh configuration, for redundancy and load balancing.

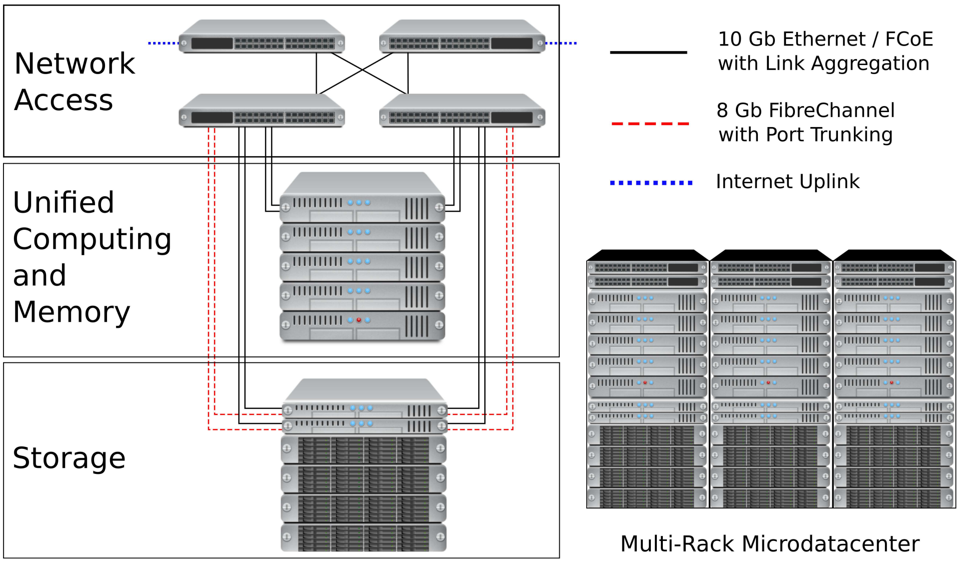

Network Access
Unified Computing and Memory
Storage
10 Gb Ethernet / FCoE with Link Aggregation
8 Gb FibreChannel with Port Trunking
Internet Uplink
Multi-Rack Microdatacenter

Figure 5.3: Generic Microdatacenter Architecture.

**Management Software:** Each vendor offers a set of management tools not only for administering the components but also to create resource slices and to delegate the operation of the slices to external entities. This can be done per-server or via hardware supported virtualization, see Chapter 2.5. The management software is also responsible for storage allocation and handling network resources, including IP address space. In addition, it comes with a monitoring interface that allows the ISP to monitor the utilization of the overall microdatacenter as well as information for each slice that can be shared with the external entity.

Figure 5.3 shows the general architecture of a Microdatacenter. An ISP can allocate resource slices consisting of computing, storage, memory, and network access in a microdatacenter and then delegate the operation of the slice to a CDI. This is what we refer to as the *ISPs cloud service* which is realized via microdatacenters slices throughout the ISPs infrastructure, which offers a large advantage over normal cloud resources: server deployment very close to the end-user.

**Definition 5.1: Microdatacenter Slice.**
A microdatacenter slice is a set of physical or virtualized resources of specified capacity within a microdatacenter. Control of the slice is delegated to the service provider that can install and operate its service using the resources of the slice.

For example, a slice can be a physical 1-core server with 2 GB RAM, 30 GB storage, 1 Gbps Internet access bandwidth, and 2 public IPs. Alternatively, it can be a virtual server with 2 GB RAM, 1 Gbps Internet access bandwidth, 1 public IP, and a pre-installed OS. With current management and virtualization tools available from established vendors, it is possible to allocate/de-allocate slices on demand with unprecedented degree of freedom, e.g., [22] and references within.

### Microdatacenter Network Footprint

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. Location diversity is highest at the edge router locations. However, it is not always possible to deploy a microdatacenter, 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. Border router locations are often a subset of core routers, hence they inherit the same advantages and disadvantages.

The main benefit of using an ISP cloud service vs. a public cloud service for a CDI is the chance to minimize the distance to the end-user. In-network server allocation allows the CDI to control the location of the slices and ensures that there are no major network bottlenecks.

## 5.2.2 Design for In-Network Server Allocation

*In-network server allocation* is a service of the ISP (see Figure 5.2) that enables CDIs to scale their infrastructure according to end-user demands, so as to minimize its capital expenditures and operating costs, as well as reducing the network distance between its infrastructure and the end-user. Moreover, it offers an interface that enables the CDI to map user requests to appropriate slices in order to maximize slice utilization and minimize the distance between the end-user and the slices.

**Definition 5.2: In-Network Server Allocation.**
The in-network server allocation is a service offered by the ISP and 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 CDI requests slices based on its clients demand. When the slice is allocated to the CDI, the service can be installed on the slice. From that point on, the CDI fully controls the operation of the service installed on the slice in the selected microdatacenter. Negotiation about microdatacenter locations and available slices are done via the

*in-network server allocation interface* through which CDI demands are matched to the ISPs resources. The interface also allows access to billing information. Moreover, the in-network server allocation interface enables the mapping of end-user requests to appropriate slices utilizing the informed user-server assignment, see Chapter 5.3

The above mentioned use of microdatacenters is in-line with the available primitives of private and public clouds operated in large-scale datacenters, e.g., [19, 117], and the recently announced Network Function Virtualisation (NFV) [123].

How to map demands to resources in an efficient manner is the task of the ISP. Only the ISP has up-to date information about the current state of the network, e.g., the internal routing configuration and the current link utilization. Thus, the ISP has to implement two basic functionalities based on the above specification to offer in-network server allocation: *mapping of CDI demands to slices* and *assigning end-users to slices*. Note, the time scales at which these two services are expected to be used differ significantly. The first one allows the service provider to flexibly allocate and de-allocate slices, e.g., based on demand forecasts. We foresee that requests for slices are not issued individually but rather collectively on a time scale of tens of minutes or hours. The second one enables ISPs to assist CDIs in assigning end-users to slices which we discuss in detail in Chapter 5.3

The CDI provides the ISP with a set of demands for slice resources, predicted demand locations, desired slice locations, as well as optimization criteria. The ISP then has to map the demands to its microdatacenter resources. We expect that the major degree of freedom that the ISP uses to jointly optimize performance is the desired slice location. We refer to this optimization problem as the *slice location* problem. If the desired slice locations are fully specified or the predicted demand locations are missing, the slice location problem becomes trivial and the ISP only grants or denies the requested slice resources.

Another degree of freedom in-network server allocation offers to the CDI is auto-scaling. While it is quite feasible to dimension applications, flash-crowds or device failures are hard to predict. To this end, in-network server allocation can offer to create replicas if its monitoring indicates that the capacity at a given location is or will be exceeded. To realize this service, the ISP needs to constantly monitor available resource and if necessary migrate or suggest the creation of additional slices. Moreover, it allows the CDI to monitor the utilization of its slices.

### Service Interfaces

The in-network server allocation of the ISP offers three interfaces to the Content Delivery Infrastructures to interact with the system:

**Resource discovery:** Using this interface the CDI requests information about resources, e.g., about available locations for slices and if in principle slices are available at those locations at what price.

**Slice allocation:** The CDI is able to requests slice allocations within a certain cost limit or with specific slice configurations via this interface.

**Monitoring and billing:** This interface allows the CDI to monitor the current status and cost of all its allocated slices.

In the remainder of this section we shortly discuss the monitoring and billing interface. In Chapter 5.2.3 we formulate the slice location problem and formulate a Linear Program for it to propose approximation heuristics based on our LP and local search. Chapter 6 gives specific examples of how the resource discovery and slice allocation interfaces can be implemented and how a CDI and ISP can utilize them to cooperate in order to improve their services and infrastructure deployments.

#### Monitoring and Billing

It is important for the CDI to minimize and track the cost of its use of in-network server allocation. Depending on the scale of the services, the service provider has to pay the usual price or negotiate bilateral agreements with the ISP. Using the resource discovery interface, it estimates the cost of slice allocation at possible locations. Using the slice allocation interface, it can bound the total cost of the request.

We expect that the billing of a slice allocated via in-network server allocation 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 licenses. The installation cost can depend on the location of the microdatacenter that hosts the slice or the time-of-day.

The usage cost follows a pay-as-you-go billing model and charges for the usage of different resources assigned to a slice. The billing among different resources in the same slice can be quite diverse. The slice can use expensive resources such as bandwidth or cheaper ones such as CPU.

For example, a slice may have a $0.01 per hour installation cost and a usage cost that depends on its use of various resources, e.g., $0.02 per real CPU usage per hour, $0.001 per GByte stored per hour, and $0.001 per Gbps outgoing traffic per hour. If the slice is idle, then only the installation cost is charged. Note, that if the slice is used for a short period within the allocation time, e.g., a few minutes, then the charge may apply to the minimum billing granularity.

To minimize the cost of deploying an on-demand service, the CDI can change its total slice demands as well as its slice specifications dynamically. Moreover, it can relax the slice specifications to reduce overall cost of its service deployment.

### 5.2.3 Algorithms for In-Network Server Allocation

The slice location problem can be modeled as an instance of the capacitated facility location problem (CFL) [92], where the locations at which facilities can be opened correspond to the locations at which slices can be placed, and there is a constraint on the amount of bandwidth available at each location or on each network link. The goal is to determine where the servers should be placed so as to satisfy all end-user demands while respecting the capacity constraints of both the slices and the underlying network infrastructure, and also possibly minimizing the distance between slices and end-users. Given the specification of a slice, if the capacity of a location allows multiple slices to be allocated then the solution may allocate more than one slice per location. As previously discussed, the ISP has a detailed view of the network activity (e.g., traffic matrices over a period of time), the annotated network topology, and the candidate locations to allocate slicea, along with the available resources, including the network capacity at these locations. The CDI can also express the demand that needs to be satisfied with additional slices as well as the slice requirements.

In the CFL solution, to prevent the creation of hot-spots, the distance of end-users to slices is proportional to the utilization of the most congested link (given the background traffic) along the path from the slice to the end-user. We also assume that the informed user-server assignment enabler is in place. In our setting end-users can be assigned to different slices for each request. Thus, the demand is in general splittable. This allows for fast and accurate server allocations using standard local search heuristics for CFL [23].

In this section we show how the slice location problem can be to formulated as a Slice Location Problem (SL) that is used to allocate slices in the ISP cloud such that the operational cost and the distance between end-users and slices is minimized. We then formulate a Linear Program for the slice location problem and propose approximation heuristics based on our LP and local search.

**The Slice Location Problem:** Let a directed graph $G = (V, E)$ represent the ISP network given by a router set $V$ and a set of links $E$. Let $L \subseteq V$ be the set of locations in the network where the ISP operates microdatacenters. Let $S$ be the set of available slices at these locations that can potentially host the service. Let $c_{ij}$ be the delay between a slice $s_i \in S$ and clients attached to $v_j$. Also, let $u_i$, $f_i$ and $r_i$ denote the slice capacity, the installation cost and the the unit price for resource utilization of slice $s_i$ respectively. Finally, let $d(v_j)$ denote the service demand of users attached to $v_j$. Let us now formally define the slice location problem:

**Definition 5.3: Slice Location Problem (SL).**
Given a set of available slices $S$ with associated installation and usage cost, and a set of demands $d(v_j)$, $\forall v_j \in V$, select a subset of slices $F \subseteq S$ so as to minimize the total cost of installing and operating the slices, as well as offering the service close to the client demand.

SL is the capacitated facility location problem (CFL) [92], where the facilities correspond to the slices and can be co-located. Moreover, to model the cost of operating a slice, the distance between a slice and clients is increased linearly by the unit price for the usage of a resource in this slice ($p_i$). We focus on the CFL with *splittable demands*, which allows demand to be allocated to more than one facility. This is a reasonable assumption as requests from different users that are attached to the same router can be served by different slices. This allows better utilization of the microdatacenter resources and thus reduces usage cost. The total number of slices that are allocated are part of the solution of the optimization problem. If $k$ is an upper bound on the number of slices that a service can install then SL is the capacitated $k$-median problem with splittable demands. We refer to this version of the slice location problem as $k$-slice location.

Both the capacitated facility location and the $k$-median problem are NP-hard [92] and therefore both versions of the slice location problem are as well. Thus, we propose heuristics based on linear programming and local search.

**Linear Programming:** We formulate SL as an integer linear program, and relax the integrality constraints to obtain a linear program (LP):

$$\begin{array}{lll}
\texttt{min} & \sum_i f_i y_i + \sum_j \sum_i r_i x_{ij} d(v_j) c_{ij} & \\
\texttt{s.t.} & \sum_i x_{ij} \geq 1 & \forall j \\
 & x_{ij} \leq y_i & \forall i, j \\
 & \sum_j x_{ij} d(v_j) \leq u_i y_i & \forall i \\
 & y_i \leq 1 & \forall i \\
 & x_{ij}, y_i \geq 0 & \forall i, j.
\end{array}$$

**Algorithm 5.1:** Linear Program for Slice Location Problem

Variable $y_i$ is boolean and indicates if a slice $s_i$ is selected ($y_i = 1$) or not. Variable $x_{ij}$ indicates the fraction of demand $d(v_j)$ that is assigned to slice $s_i$. The first constraint states that the demand has to be satisfied and the second one that the demand can be assigned only to selected slices. The third constraint states that the total demand served by a slice can not exceed the slice capacity. The fourth constraint states that a slice can be served only once and the last one that no negative fraction of demand can be assigned or no unavailable slice can be assigned. To solve the $k$-slice location, one more constraint must be added: $\sum_i y_i \leq k$.

The solution of the above LP can be found in polynomial time [88]. A number of techniques have been proposed to find a solution faster and include rounding [39,104] and primal-dual methods [82]. The above LP can be extended to tackle the resource requests of multiple CDIs at the same time.

**Fast Heuristic – Local Search:** The LP solution may be too slow for slice location as its run-time scales with the number of microdatacenters and CDIs. Therefore, we consider alternative heuristics. The best heuristic for the facility location and $k$-median problems is *local search* [90]. In the setting of the slice location problem, the local search heuristic starts with an initial feasible allocation of slices. Then, it incrementally improves the solution either by evaluating neighboring solutions, e.g., by adding, removing, or swapping one or more slices. Once the local search finds a stable set of slices it has found its local optimum. For the slice location problem with splittable demands, a local search heuristic that permits adding, dropping, or swapping one slice has been shown to give good approximations [23].

## 5.3 Informed User-Server Assignment

The need for informed user-server assignment is motivated by the observation (see Figure 5.1) that by selecting an appropriate CDI server out of all the available ones (servers A, B, C), it is possible to improve end-user performance and at the same time achieve traffic engineering goals. Today, the massive deployment of CDI servers offers both server and path diversity that is largely unexplored. The latter is due to the fact that CDIs and ISPs operate in isolation.

### 5.3.1 Design for Informed User-Server Assignment

As pointed out ISPs are in a unique position to help CDIs and P2P systems to improve content delivery since they have the knowledge about the current state of the underlying network topology, the status of individual links, as well as the precise network location of the end-user. The idea to leverage this information ISPs have readily at hand is by no means new and thus the research community has proposed various alternative solutions. While they all differ in certain aspects, their basic idea is the same: utilize available information about the network to make an educated selection prior connecting to a service. Following this idea, all of the proposed solution employ the same basic conceptual design: the *management plane* is responsible for collecting up-to-date information about the network while the *control plane* acts as an interface to this information for the application. In this section we formally introduce the generic design for informed user-server assignment and present readily available systems and how they realize this design in detail.

**Management Plane – The Network Map:** The systems management plane is responsible to collect up-to-date state network information, such as network topology, routing information, link utilization and other important metrics. This information is used to maintain an internal map of the network representing the current state of the real network. One important aspect of this component is how the information about the network is retrieved. The different implementations range from active measurements over passive measurements to active participation in network management systems (such as BGP). Another important aspect is the frequency in which the information is collected. For certain information such as topology or routing an immediate update is necessary to guarantee correct functioning of the system, while others, such as link utilization or packet loss rates, only degrade the quality of the system. Still other information, such as link capacities or transmission delays, can be considered (semi-)static. Last but not least the systems differ in what information is necessary to be operational and if additional information sources can be used to improve accuracy.

**Control Plane – The Information Interface:** The control plane of the system is responsible for providing an interface to the information of the management plane so that clients can make use of the information. This can basically be seen as an interface or API that clients can query to get information about the current network state. The various proposed solutions differ mainly in which fashion and at which granularity the information can be retrieved. There are two main competing approaches: abstracted network maps and preference lists. The first one transforms the available information from the management plane into an annotated representation of nodes and edges. The big difference to the actual data of the management plane is the aggregation level and the specific annotations. Clients can then query the system to get an up-to-date abstract network map, which they can use to decide which of the possible destination to connect to by calculating the best candidates by themselves using their own optimization target. The second one uses the information of the management plane to create a ranked list of possible service destinations (read: IP addresses). The required input includes the source, possible destinations and (if the system supports multiple collaboration objectives) an optimization goal, e.g., minimal delay. The output consists of a re-ordered list of the possible destinations in regard to the

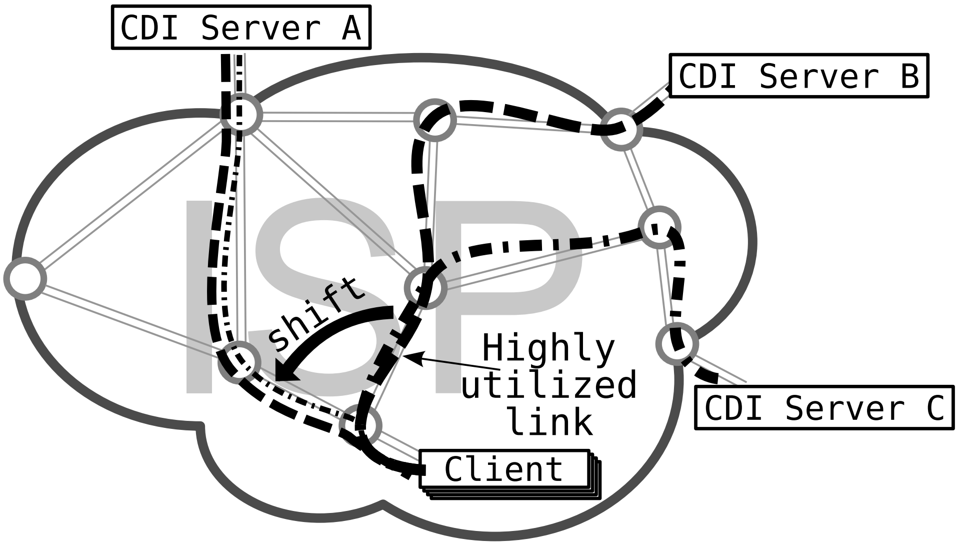


Figure 5.4: Informed User-Server Assignment Process

optimization goal, the first being the most and the last being the least desirable destination.

Note that in both cases the client is in the position to select the final destination, allowing to completely ignore the additional information. Another important fact is that the client is not necessarily the end-user but can be a service provider themselves. For instance a company providing content delivery service (CDI) can make use of this service to improve its user-to-server mapping accuracy or in case of the BitTorrent P2P system the tracker can query the service prior returning an initial peer list to a connected client. While not strictly necessary, the two components are usually implemented as separate entities within the system to allow better scalability, information aggregation and/or anonymization without loosing precision or multiple collaboration objectives. In addition to that, all systems tackle important issues for any collaboration approach, such as privacy information leakage or targeted objective(s).

Figure 5.4 illustrates how those systems can influence the traffic flows inside a network. With the recommendation from the ISP the server selection process of a CDI avoids the highly utilized link and thus improves the end-users performance while at the same time the ISP is able to better balance the traffic inside the network.

### Proposed Solutions for Informed User-Server Assignment

In this section we introduce the different solutions proposed by the research community for informed user-server assignment and outline the specific implementation and thus highlights the differences between them. The presented solutions include the original Oracle concept proposed by Aggarwal et al. [10], P4P proposed by Xie et al. [177], Ono proposed by Choffnes and Bustamante [41] and PaDIS proposed by Poese et al. [138]. We also give an overview of the activities within the IETF Application Layer Traffic Optimization (ALTO) working group, which have been fueled to some extend by the proposed systems discussed in this section.

**P2P Oracle Service:** Aggarwal et al. [10] describe an *oracle* service to solve the mismatch between the overlay network and underlay routing network in P2P content delivery. Instead of the P2P node choosing neighbors independently, the ISP can offer a service, the *oracle*, that ranks the potential neighbors according to certain metrics: a client supplied peer list is re-ordered based on coarse-grained distance metrics, e.g., the number of AS hops [79], the peer being inside/outside the AS or the distance to the edge of the AS. This ranking can be seen as the ISP expressing preference for certain P2P neighbors. For peers inside the network additional information can be used, such as access bandwidth, expected delay or link congestion to further improve the traffic management.

**Proactive Network Provider Participation for P2P (P4P):** The "Proactive Network Provider Participation for P2P" is another approach to enable cooperative network information sharing between the network provider and applications. The P4P architecture [177] introduces iTrackers as portals operated by network providers that divides the traffic control responsibilities between providers and applications. Each iTracker maintains an internal representation of the network in the form of nodes and annotated edges. A node represents a set of clients that can be aggregated at different levels, e.g., certain locations (PoP) or network state (similar level of congestion). Clients can query the iTracker to obtain the "virtual" cost for possible peer candidates. This "virtual" cost allows the network operators to express any kind of preferences and may be based on the provider's choice of metrics, including utilization, transit costs, or geography. It also enables the client to compare and choose the most suited peers to connect to.

**Ono - Travelocity-based Path Selection:** The Ono system [41] by Choffnes and Bustamante is based on "techniques for inferring and exploiting the network measurements performed by CDNs for the purpose of locating and utilizing quality Internet paths without performing extensive path probing or monitoring" proposed by Su et al. in [165]. Based on their observations that CDN redirection is driven primarily by latency [165], they formulate the following hypothesis: Peers that exhibit similar redirection behavior of the same CDN are most likely close to each other, probably even in the same AS. For this each peer performs periodic DNS lookups on popular CDN names and calculates how close other peers are by determining the cosine similarity with their lookups. To share the lookup among the peers they use either direct communication between Ono enabled peers or via distributed storage solutions e.g., DHT-based. On the downside Ono relies on the precision of the measurements that the CDNs perform and that their assignment strategy is actually based mainly on delay. In case the CDNs change their strategy in that regard Ono can yield wrong input for the biased peer selection the authors envision.

When considering our design concept described above, Ono is a bit harder to fit into the picture: Ono distributes the functionality of the management and control planes among all participating peers. Also, Ono does not try to measure the network state directly, but infers it by observing Akamai's user-to-server mapping behavior on a large scale and relies on Akamai doing the actual measurements [128], Thus the management plane of Ono consists of recently resolved hostnames from many P2P clients. The quality of other peers can then be assessed by the number of hostnames that resolve to the same destination. The control plane in Ono's case is a DHT, which allows decentralized reads and writes of key-value pairs in a distributed manner, thus giving access to the data of the management plane.

**Provider-aided Distance Information System (PaDIS):** In [138] Poese et al. propose a "Provider-aided Information Systems (PaDIS)", a system to enable collaboration between network operators and content delivery systems. The system enhances concept of the P2P Oracle to include server based content delivery systems (e.g., CDNs), to maintain an up-to-date annotated map of the ISP network and its properties as well as the state of ISP-operated servers that are open for rent. In addition, it provides recommendations on possible locations for servers to better satisfy the demand by the CDN and ISP traffic engineering goals. In the management plane, it gathers detailed information about the network topology, i.e., routers and links, annotations such as link utilization, router load as well as topological changes. An Interior Gateway Protocol (IGP) listener provides up-to-date information about routers and links. Additional information, e.g., link utilization and other metrics can be retrieved via SNMP. A Border Gateway Protocol (BGP) listener collects routing information to calculate the paths that traffic takes through the network, including egress traffic. Ingress points of traffic can be found by utilizing Netflow data. This allows for complete forward and reverse path mapping inside the ISP and enables a complete path map between any two points in the ISP network. While PaDIS builds an anotated map of the ISP network, it keeps the information acquired from other components in separate data structures. This separation ensures that changes in prefix assignments do not directly affect the routing in the annotated network map. Pre-calculating path properties for all paths, allow for constant lookup speed independent of path length and network topology. On the control plane, PaDIS makes use of the prefrence lists known from the P2P Oracle, but supports multiple, individual optimization targets. Apart from basic default optimizations (e.g., low delay, high throughput), additional optimizations can be negotiated between the network operator and the content delivery system.

**Application-Layer Traffic Optimization (ALTO):** The research into P2P traffic localization has led the IETF to form a working group for "Application Layer Traffic Optimization (ALTO)" [113]. The goal of the ALTO WG is to develop Internet standards that offer "better-than-random" peer selection by providing information about the underlying network and to design a query-response protocol that the applications can query for an optimized peer selection strategy [18]. On the control plane, ALTO offers multiple services to the applications querying it, most notably are the Endpoint Cost Service and the Map service. The Endpoint Cost Service allows the Application the query the ALTO server for costs and rankings based on endpoints (usually IP subnets) and use that information for an optimized peer selection process or to pick the most suitable server of a CDI. The Network Map service makes use of the fact that most endpoints are in fact rather close to each other and thus can be aggregated into a single entity. The resulting set of entities is then called an ALTO Network Map. The definition of proximity in that case depends on the aggregation level, in one Map endpoints in the same IP subnet may be considered close while in another all subnets attached to the same Point of Presence (PoP) are close. In contrast to the Endpoint Cost Service the ALTO Network Map is suitable when more Endpoints need to be considered and offers better scalability, especially when coupled with caching techniques. Although the ALTO WG statement is more P2P centric, the service is also suitable to improve the connection to CDN servers.

### 5.3.2 Algorithms for Informed User-Server Assignment

In this section we propose algorithms to realize informed user-server assignment in the context of an ISP. A key observation is that informed user-server assignment can be reduced to the restricted machine load balancing problem [27] for which optimal online algorithms are available. The benefit of the informed user-server assignment 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 informed user-server assignment offers to an ISP, we present offline algorithms that uses traffic demands and server diversity over time extracted from those statistics as input.

For this, the CDI identifies the servers that can satisfy the requested demand and then ranks them based on its own criteria. Let $S' \subseteq S$ be the set of possible CDI servers and $\{x_i\}$ and $\{y_i\}$, $i \in [1, |S'|]$ a ranking of the servers from the viewpoint of the CDI and ISP, respectively. The CDI then assigns the user to the slice that minimizes the rankings of the CDI and the ISP. We formally define the joint optimization problem of assigning end-users to servers or slices hosted in ISP microdatacenters as informed user-server assignment problem:

**Definition 5.4: User-Server Assignment Problem.**
Given a new demand request $d_r$ from an end-user that originates at $v_j$, and a set of servers $S' \subseteq S$ that can satisfy the request, assign the end-user to the server $s_i \in S'$

that optimizes the CDIs criteria, e.g., minimum cost, while considering the ISP's recommendations.

### Connection to Restricted Machine Load Balancing

Given a set of CDIs and their network location diversity, we consider the problem of re-assigning the flows that correspond to demands of end-users to the CDIs in such a way that a specific optimization goal is achieved. Given that sub-flows between end-systems and CDI servers can be re-distributed only to a subset of the network paths, we show that the solution of the optimal traffic matrix problem corresponds to solving the *restricted machine load balancing problem* [27]. In the restricted machine load balancing problem, a sequence of tasks is arriving, 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 between only one source-destination pair are utilized, informed user-server assignment can utilize any eligible path between any candidate source and destination of traffic.

For ease of presentation let us assume that the optimization goal is to minimize the maximum link utilization in the network [64, 65]. Let us consider three consumers where each one wants to download one unit of content from two different content delivery infrastructures, see Figure 5.5. Given that different servers can deliver the content on behalf of the two CDIs, the problem consists in assigning end-users 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 one 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 5.5 shows one of the possible solutions to this problem, where end-user 1 is assigned to servers 1 and 4, end-user 2 to servers 5 and 2, and end-user 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 the end-users in the network, $K$ be the set of content delivery infrastructures, and $I$ be the set of servers for a given CDI, 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 end-user $j$.

**Definition 5.5: Restricted Flow Load Balancing Problem.**
The restricted flow load balancing problem is the problem of finding a feasible assignment of flows such 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 delivery infrastructure $k \in K$ to a end-user $j \in J$, and a set of eligible residual flows $f_{ij}^{-k}$, $i \in M_{jk}$ (after removing the traffic of the above mentioned sub-flows).

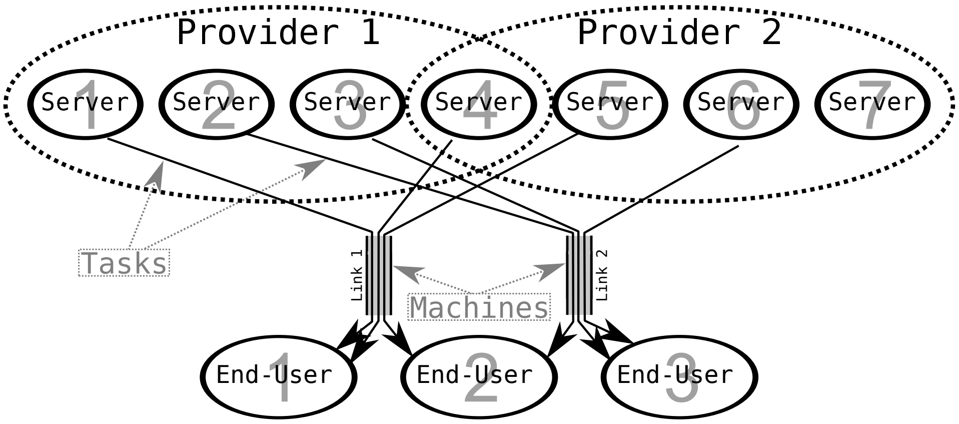


Figure 5.5: User-Server Assignment and Restricted Machine Load Balancing.

Despite some similarities, the nature of our problem differs from the multi-commodity flow and bin packing. In the multi-commodity flow problem [25], 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 [45], 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 a destination server can be used, contrary to the multipath problem where only equal-cost paths can be used.

### 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 [27], 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 an online one, thus it converges to the optimal solution immediately without oscillations.

In the restricted flow load balancing problem, the set $M_{jk}$ can be obtained from the set of candidate servers that can deliver content when utilizing informed user-server assignment as described in Chapter 5.3. The online assignment of users to servers per request, which minimizes the overall load, leads to an optimal assignment of sessions within sub-flows. In our case, flows are splittable since the content corresponding to each content request is negligible compared to the overall traffic traversing a link. Note, the end-to-end TCP connections are not splittable. Thus, the following online algorithm is optimal:

**Algorithm 5.2: Online Greedy Server Selection.**
Upon the arrival of a content request, assign the end-user to the server that can deliver the content, out of all the servers offered by the CDI, such that the traffic engineering goal is achieved.

### Offline Algorithm

Before applying informed user-server assignment in real operational networks, it is important to understand the potential benefits that it can bring in a given context. For example, the operator of an ISP network wants to know in advance what are the gains when applying informed user-server assignment, as well as being able to answer what-if scenarios, when applying informed user-server assignment to traffic delivered by different CDIs. Companies operating CDIs also want to quantify the benefits by participating in informed user-server assignment before collaborating with an ISP. In most operational networks, aggregated statistics and passive measurements are collected to support operational decisions. Therefore, we provide a framework that allows a simulation-driven evaluation of informed user-server assignment in Chapter 6. To that end, we now present offline algorithms that can take as input passive measurements and evaluate the potential gain when applying informed user-server assignment in different scenarios. We propose a linear programming formulation as well as greedy approximation algorithms to speed-up the process of estimating the gain when using informed user-server assignment.

**Linear Programming Formulation:** To estimate improvement of informed user-server assignment we formulate the Restricted Flow Load Balancing problem (see Chapter 5.3.2) 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 end-user $j$ cannot download the content from server $i$ of a content provider $k$. For each end-user $j$ 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. For ease of presentation we use as objective function the minimization of the maximum link utilization. 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:

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 [88]. The solution is the optimal flow assignment, $f^*_{ijk}$, that minimizes the maximum link utilization of the network. If flows are not splittable, or the sub-flows are discretized,

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

**Algorithm 5.3:** Linear Program for Informed User-Server Assignment

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 [103].

**Approximation Algorithms:** Since it is a common practice for operators to study multiple scenarios to quantify the effect of changes in traffic matrices over periods that spans multiple weeks or months, solutions based on LP may be too slow. It is also too slow to estimate the gain of informed user-server assignment when applying it to an arbitrary combination of CPs. To that end, we turn our attention to the design of fast approximation algorithms. Simple greedy algorithms for load balancing problems [77] are among the best known. Accordingly, we propose a greedy algorithm for our problem which starts with the largest flow first.

**Algorithm 5.4: Greedy-Sort-Flow.**
Sort sub-flows in decreasing order based on volume and re-assign them in this order to any other eligible flow which, after assigning the sub-flow $f_{ijk}$, will yield the most for the desired traffic engineering goal.

Assignment in sorted order has been shown to significantly improve the approximation ratio and the convergence speed [51,77]. Recent studies [71,98,137] show that a small number of content delivery infrastructures are responsible for a large fraction of the traffic. Therefore it is expected that the algorithm yields results close to the optimal ones. To further improve the accuracy of the proposed approximation algorithm, we design an *iterative* version of the algorithm, presented in Algorithm 5.5, that converges to the optimal solution. Indeed, a small number of iterations, typically one, suffice to provide a stable assignment of flows.

**Algorithm 5.5:** 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 re-assigned 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.
      ▷ Re-assign $f_{ijk}$ in $f_{ij}^{-k}$, $i \in M_{jk}$ s.t. the engineering goal is achieved.
```

## 5.4 Summary

In this Chapter, we introduce two key enablers, namely in-network server allocation and informed user-server assignment, to facilitate the collaboration between CDIs and ISPs and to address the challenges of todays content delivery landscape discussed in Chapter 3.1. We describe the two enablers in great detail by outlining their design rationale and propose efficient algorithms to realize them. We find that leveraging the information that both the CDI and the ISP have readily at hand enables a collaborative approach for jointly optimizing the efficiency and performance of allocating additional server resources and assigning end-user to servers. We present the system architecture of our approach, called NetPaaS, in Chapter 6 and quantify the possible benefits of the system in Chapter 7.

# 6 NetPaaS - Network Platform as a Service

Today there is no system to support CDI-ISP collaboration and joint CDI server deployment within an ISP network. In this Chapter, we present the architecture of a novel system, NetPaaS (Network Platform as a Service), which realizes the two key enablers for CDI-ISP collaboration introduced in Chapter 5. NetPaaS orchestrates the on-demand deployment of services inside microdatacenters by utilizing the view of the ISP about the network and additional computation and storage resources inside the network. First, we give an overview of NetPaaS and describe its functionalities and the protocols it utilizes to enable collaboration. Next, we give a detailed description of the NetPaaS architecture. Finally we comment on the scalability and privacy preserving properties of NetPaaS and discuss a possible deployment scenario of the system inside an ISP.

## 6.1 NetPaaS Functionalities and Protocols

NetPaaS enables CDIs and ISPs to efficiently coordinate the user to server assignment and allows the CDI to expand or shrink its footprint inside the ISPs network on demand, towards achieving performance targets [96] and traffic engineering goals [140]. Neither of them is a trivial task when dealing with large networks (thousands of routers), highly distributed microdatacenters (in tens of locations and hundreds of machines), and constant network, routing, and traffic updates.

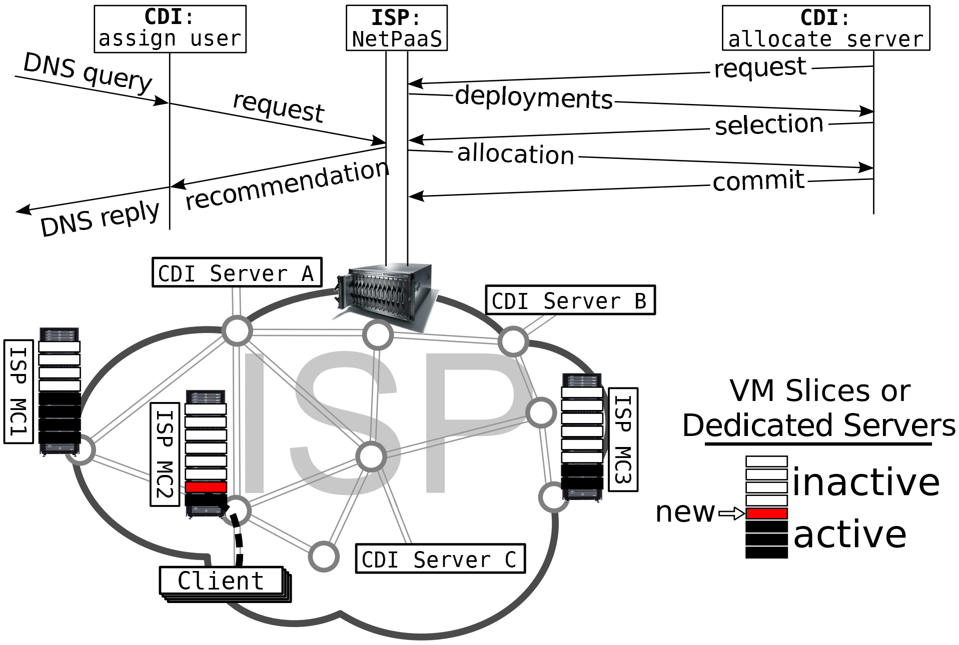


Figure 6.1: NetPaaS protocols and operation.

The NetPaaS protocol allows CDIs to express required server specifications and ISPs to communicate available resources and their prices. Its design allows the parties to exchange information at very small time scales, e.g., in the order of seconds, similar to the time scale that CDIs can potentially redirect users (see Chapter 4.7) to enable fast responses to rapid changes in traffic volumes. With NetPaaS an ISP can offer the following services: ***(1)*** **User-server assignment**: allows to request recommendations for user to server mapping from the ISP. ***(2)*** **Resource discovery**: communicates information about resources, e.g., available locations or number of servers and the conditions for leasing them, e.g., price and reservation times. ***(3)*** **Server allocation**: enables a CDI to allocate server resources within the ISPs network.

NetPaaS protocols are designed to be efficient and to minimize delay and communication overhead. The required communication for the different services are explained in more detail in Chapter 6.1.1 and 6.1.2. For the informed user-server assignment service NetPaaS also supports BGP as communication protocol as this is already supported by many CDI operators, e.g., Google Global Cache [75], Netflix Open Connect [122], or the Akamai Network [128].

### 6.1.1 NetPaaS Protocol for User-Server Assignment

We first describe the general approach for informed user-server assignment today and then discuss the additional steps and protocol messages for our collaborative approach, illustrated in the top left of Figure 6.1 ("CDI: user assign"). When a CDI receives a DNS request, typically by a resolver (i.e., when the answer is not locally available in the local resolver), it utilizes internal information in order to assign a server to satisfy the request. The selection of the server depends on the location of the source of the request, as this is inferred from the resolvers that sends it, as well as the availability of close-by servers and cost of delivery [128, 165]. When the CDI selects a set of servers to satisfy the request, it sends a DNS reply back to the resolver that sent the DNS request who then sends it to the source of the request. Notice that for scalability reasons and to deal with flash crowds, large CDIs allow all the available servers to serve the same content [169]. If the content is not locally available, the server fetches the content from other close-by servers or the origin server, caches it locally and sends it to the end-user [128]. To take advantage of the ISPs NetPaaS informed user-server assignment service the CDI issues a recommendation request prior to answering the DNS query. The recommendation request contains the source of the DNS request and a list of eligible CDI server IPs which NetPaaS ranks based on ISP-internal information, e.g., link utilization or path delay, and possible traffic engineering goals. If the source of the DNS request is the ISP operated DNS resolver or when the EDNS0 Client Subnet Extension [47] is present, NetPaaS can precisely locate the end-user inside the ISPs network, effectively increasing the recommendations precision of the system. The ISP then returns this preference ordered list in a recommendation message to the CDI which can select the most appropriate servers based on both the ISPs and its own criteria. Thus, it can optimize the informed user-server assignment while completely controlling of the final server selection process.

### 6.1.2 NetPaaS Protocol for Server Allocation

We next describe the steps and required protocol messages for collaborative in-network server allocation that are illustrated in the top right of Figure 6.1 ("CDI: allocate server"). When a CDI decides that additional server resources are needed to satisfy the end-user demand or when the CDI and ISP jointly agree to deploy new servers inside the ISP, the CDI submits a request to NetPaaS. The request contains the required hardware resources, a demand forecast (e.g., per region or per sub-net) together with a number of optimization criteria and possible constraints. The demand forecast allows NetPaaS to compute an optimal placement for the newly allocated server(s). Optimization criteria include minimizing network distance or deployment cost among others. Possible constraints are the number of locations, minimum resources per server, or reservation time. Based on this information NetPaaS computes a set of deployments, i.e., the server locations and the number of servers, by solving an optimization problem (namely the CFL problem, see Chapter 5.2.3). The reply contains the possible deployments and their respective prices. The CDI either selects one or more of the offered deployments by sending a selection message to NetPaaS or starts over by submitting a new request. When receiving a selection message, NetPaaS checks if it can offer the selected resources. If all conditions are met, NetPaaS reserves the requested resources to guarantee their availability and sends an allocation message as confirmation to the CDI. If the conditions cannot be met, the selection by the CDI is denied by NetPaaS. To gain control of the allocated servers, the CDI has to send a commit message to NetPaaS which completes the communication for in-network server allocation.

The ISP may offer physical machines or virtual machines (VMs) to CDIs. In the second case the servers are referred to as "slices" of Microdatacenters. To move servers from one to another network position, NetPaaS supports the flexibility of VM migration or consolidation. A possible deployment scenario with VMs can be seen in Figure 6.1. Here, an end-user inside the ISP is redirected to the newly allocated microdatacenter slice in MC2 instead of either one of the three available CDI servers A, B or C. To improve CDI server start-up and cache warm-up times, one option for CDIs is to always keep a small number of active servers in a diverse set of locations to expand or shrink it according to the demand. They can also pre-install an image of their server in a number of locations.

## 6.2 Architecture

We now describe the detailed architecture of the system, which provides accurate user-server assignments as well in-network server allocations for the CDI. We discuss the components and processes both at the ISP as well as the CDI side. In the ISP the main tasks of our system are to: (1) maintain an up-to-date annotated map of the ISP network and its properties as well as the state of the ISP-operated servers within the network, (2) provide recommendation on where servers can be located to better satisfy the demand by the CDI and ISP traffic engineering goals, and (3) to assist the CDI in informed user-server assignment and in-network server allocation by creating preference rankings based on the current network conditions. The goal of the system is to fully utilize the available server and path diversity as well as ISP-maintained resources within the network, while keeping the overhead for both the CDI and the ISP as small as possible.

NetPaaS comprises three main components: *Network Monitoring*, *Informed User Assignment*, and *Server Allocation Interface*. For an overview of the architecture, see the ISP gray area in Figure 6.2. Steps 1-10, I-IV illustrate the requests and responses and the CDI server selection respectively, as performed in currently deployed CDIs, for more information and details see [128]. The additional steps for integrating NetPaaS, steps A-D for informed user-server assignment and V-VIII for in-network server allocation are described in the respective component description.

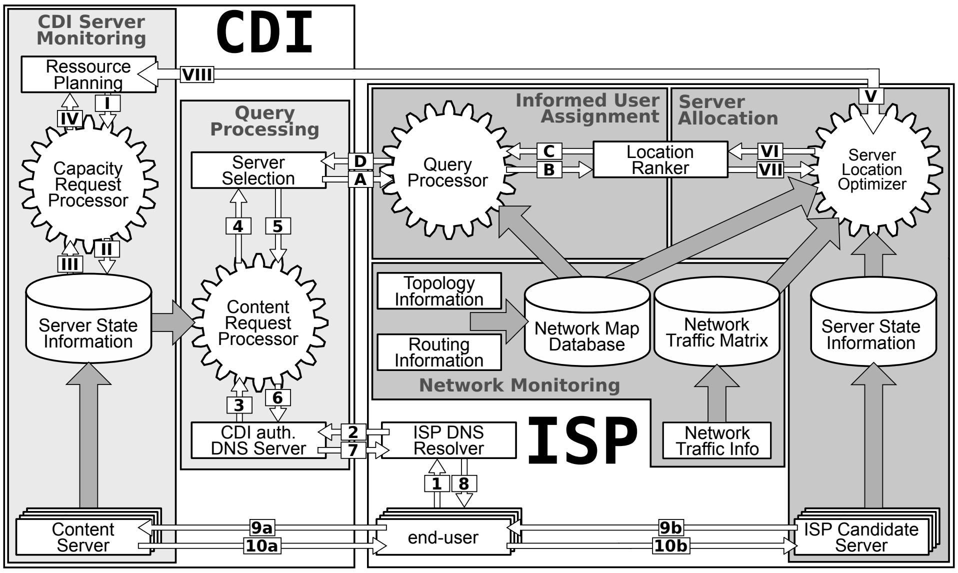


Figure 6.2: NetPaaS architecture.

### 6.2.1 Network Monitoring

The Network Monitoring gathers information about the topology and the state of the network to maintain an up-to-date view of the network. The Network Traffic Information component gathers detailed network traffic statistics in form of traffic matrices. Traffic matrices represent the traffic volumes between all possible pairs of source and destination routers in a specific time interval, e.g., per hour or per day. The Network Traffic Matrix Database stores these matrices to enable demand forecasts and traffic analytics based on the current and historic traffic matrices. The Topology Information component gathers detailed information about the network topology, i.e., routers and links, annotations such as link utilization, router load as well as topological changes. An Interior Gateway Protocol (IGP) listener provides up-to-date information about routers and links. Additional information, e.g., link utilization and other metrics can be retrieved via SNMP from the routers or an SNMP aggregator. The Routing Information component uses routing information to calculate the paths that traffic takes through the network. Finding the path of egress traffic can be done by using a Border Gateway Protocol (BGP) listener. Ingress points of traffic into the ISP network can be found by utilizing Netflow data. This allows for complete forward and reverse path mapping inside the ISP and therefore enables path lookups between any two points in the ISP network. The Network Map Database processes the information collected by the Topology and Routing Information components to build an annotated map of the ISP network. While it builds its map of the network, it keeps the information acquired from the other two components in separate data structures. The Topology Information is stored as a weighted directed graph, while the prefix information is stored in a Patricia trie [119]. This separation ensures that changes in prefix assignment learned via BGP do not directly affect the routing in the annotated network map. To further improve performance, the path properties for all paths are pre-calculated. This allows for constant lookup speed independent of path length and network topology. Having ISP-centric information ready for fast access in a database ensures timely responses and high query throughput of the NetPaaS system.

### 6.2.2 Informed User-Server Assignment

When the CDI sends a request for informed user-server assignment to NetPaaS, the request is handled by the Query Processor (steps A to D in Figure 6.2) as follows: The request from the CDI (A) specifies the end-user and a list of candidate CDI servers. First, the Query Processor maps each source-destination (server to end-user) pair to a path in the network. Note that the end-user is usually seen through its DNS resolver, often the ISPs DNS resolver [6], unless both ISP and CDI support the EDNS0 Client Subnet Extension [47, 132]. The properties of the path are then retrieved from the Network Map Database. Next, the pairs are run individually through the Location Ranker subcomponent (step B) to get a preference value (step C). Finally, the list is sorted by preference values, the values stripped from the list, and the list is sent back to the CDI (step D). The ISP Location Ranker computes the preference value for individual source-destination pairs based on the path properties and an appropriate function (see steps B and C). The function depends on the goal specified by the CDI, such as a performance goal, as well as an operational one, such as a traffic engineering objective (see Chapter 5.2.3). Note that NetPaaS is not limited to a single optimization function per CDI but enables the use of multiple, individually tweaked functions for each CDI to realize the desired optimization.

### 6.2.3 In-network Server Allocation

When the CDI Resource Planning observes capacity shortages (steps I-IV in Figure 6.2), it sends a in-network server allocation request to NetPaaS asking for available servers within the ISP (steps V to VIII). The request is handled by the ISP Server Location Optimizer (step V) and contains possible constraints and requirements, such as optimization criteria, slice specifications, favored locations, and demand forecasts. It uses the Network Monitoring component to get up-to-date information about the ISPs network, the current and historic network traffic matrices, and the Server State Information database, which collects up-to-date state information regarding the ISP's servers (e.g., server load and connectivity). Each microdatacenter slice that meets the requirements and constraints is then fed to the ISP Location Ranker (step VI) which solves the Slice Location problem (see Chapter 5.3.2) for all eligible slices based on the network path properties and an appropriate optimization function. The outcome of the joint server allocation (step VII) is the number and location of additional servers and the result is communicated back to the CDI (step VIII) which then decides to agree on the calculated setting or restart the in-network server allocation process.

**Joint Hardware Server Allocation:** Here, the collaboration of the ISP and CDI is at large time scales in the order of days or weeks and the servers are physical machines installed and maintained by the ISP and operated by the CDI. In the setting of the ISP-operated CDI, the in-network server allocation is an optimized way of deploying the CDI footprint inside the network. The forecast of the demand by analyzing CDI logs can also be incorporated. This joint operation also allows the launch of new and demanding applications such as video streaming and interactive online gaming.

**Joint Software Server Allocation:** As mentioned before, servers can be either physical machines owned by the CDI, virtual machines offered by the ISP, or both. With virtualization, the above solution can be utilized whenever software servers are allocated. This allows for flexible server allocation using a mature technology. Virtualization has been used to allocate heterogeneous resources [171, 175], computation (e.g., VMWare, Xen, and Linux VServer), storage, and network [153], in datacenters [22], as well as distributed clouds inside the network [42, 123]. Recent measurement studies have shown significant performance and cost variations across different virtualization solutions [105]. In response, a number of proposals have addressed the specific requirements of applications [29, 93, 110] and the scalability to demand [142, 176]. To capitalize on the flexibility and elasticity offered by virtualization, a number of systems have been built to automate data and server placement [4, 50, 174] and server migration [35, 101] even between geographically distributed datacenters. Other approaches have focused on the selection of locations for service mirrors and caches inside a network, to minimize the network utilization [95, 100]. In the joint server allocation setting the decision and installation time can be reduced to hours or even minutes. This is feasible as an ISP can collect near real-time data for both the network activity and availability of resources in datacenters operated within its network or in microdatacenters collocated with ISP network aggregation points [42].

## 6.3 Scalability

**User-Server Assignment:** To improve scalability and responsiveness, we do not rely on HTTP embedded JSON as proposed in by ALTO IETF group, but on light protocols that are similar to DNS. A single instance of our system is able to reply to more than 90, 000 queries/sec when serving requests with 50 candidate CDI servers. At this level, the performance of our system is comparable to popular DNS servers, e.g., BIND. The computational response time is below 1 ms for a 50 candidate server list. By placing the service inside ISP networks at well connected points, the additional overhead is small compared to the DNS resolution time [6]. This performance was achieved on a commodity dual-Xeon CPU (8 cores, 2.5 GHz) server with 16 GByte RAM and 1 Gbps Ethernet interfaces. Furthermore, running additional servers does not require any synchronization between them since each instance is acquiring the information directly from the network. Thus, multiple servers can be located in different places inside the network to improve scalability.

**Server Allocation:** Today, a number of off-the-shelf solutions are available to spin a virtual server based on detailed requirements [110], and are already available from vendors such as NetApp and Dell. To test the scalability of in-network server allocation we used an appliance collocated with a network aggregation point of ADSL users which consists of 8 Xeon CPUs (48 cores, 3 GHz), 96 GByte RAM, multiple Terabytes of solid state disks, and a 10 Gbps network interface. A management tool that follows the VMware, Cisco, and EMC (VCE) consortium industrial standard [171] is also installed. We tested different server configurations and our results show that VM boot up times are on the order of tens of seconds while virtualization overhead during runtime is negligible. To that end we confirm that it is possible to even fully saturate a 10 Gbps link. It was also possible to add, remove, and migrate live servers on demand in less than a minute. To reduce the cache warm-up time when allocating a new server, the requests to an already operational cache are duplicated and fed to the new one for around ten minutes.

## 6.4 Privacy

During the exchange of messages, none of the parties is revealing sensitive operational information. In informed user-server assignment, CDIs only reveal the candidate servers that can respond to a given request without any additional operational information (e.g., CDI server load, cost of delivery). The ISPs do not reveal any operational information or the preference weights they use for the ranking. In fact, the ISPs only re-order a list of candidate servers provided by the CDI. This approach differs from [177], where partial or complete ISP network information, routing weights, or ranking scores are publicly available. During the in-network server allocation a CDI can decide either to request a total or regional (e.g., city, country) demand, thus it does not unveil the demand of an end-user. We believe that the final decision must still be made by the CDI, yet be augmented with up-to-date network guidance from the ISP.

## 6.5 Deployment

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

Typically, an ISP operates a number of DNS resolvers to better balance the load of DNS requests and to locate DNS servers closer to end-users. To this end, we envision that the ISP's NetPaaS servers can be co-located with DNS resolvers in order to scale in the same fashion as DNS. NetPaaS servers can also be located close to peering points in order to reduce the latency between the CDI and an instance of the system. Synchronization of multiple NetPaaS instances is not necessary as it is implicitly given through the use of protocol listeners and queries regarding available system resources on demand.

## 6.6 Summary

In this Chapter we present the architecture of NetPaaS, a novel system to orchestrate joint on-demand deployment of CDI server resources. The system enables CDI-ISP collaboration by leveraging the view of the ISP about the network and available microdatacenter resources inside the ISPs network. We describe the necessary functionalities and protocols to realize NetPaaS. Our architecture describes the essential system components, namely the network monitoring, the informed user assignment and the server allocation. For each component, we outline the necessary means to gather available information and describe how the CDI interacts with NetPaaS to take advantage of the system. Last but not least, we comment on the scalability of the system, discuss the privacy preserving properties of NetPaaS, and present a possible deployment scenario of the system inside an ISP. In the next Chapter, we evaluate NetPaaS and quantify possible benefits when a large European Tier-1 ISP collaborates with the largest commercial CDI to jointly deploy additional content delivery infrastructure.

# 7 NetPaaS Evaluation

In this chapter we quantify the benefits of using NetPaaS. For our evaluation we rely on traces from the largest commercial CDI and a European Tier-1 ISP. We start by introducing the utilized simulation environment and the used datasets. Next, we present the traffic characteristics of the CDI inside the ISP and analyze the collaboration potential of NetPaaS. We continue by evaluating the possible benefits and improvements of NetPaaS regarding relevant network metrics, such as delay and network wide traffic. To this end, we consider multiple different optimization goals, e.g., reducing the network wide traffic, optimizing the delay between the end-user and the CDI server, and reducing the maximum link utilization. We start by quantifying the improvements of informed user-server assignment and continue with the benefits of in-network server allocation. We conclude our evaluation by anticipating the launch of a traffic intensive service exclusively utilizing NetPaaS and the implications of the ISP collaborating with multiple CDIs.

## 7.1 Simulation Environment

To evaluate the potential benefits of NetPaaS we rely on a simulator. Our simulator takes as input (i) the annotated topology and routing information for the considered ISP, (ii) the traffic demands of the largest commercial CDI, (iii) an optimization goal, e.g., optimizing the delay between end-users and CDI servers (iv) the traffic demands of additional CDIs, (v) possible locations for microdatacenter slices inside the ISP network, and (vi) traffic matrices representing the traffic inside the ISP without the load imposed by the CDIs – the background traffic. Based on this input, the simulator computes the resulting network loads inside the ISP network by computing solutions to the slice location (see Chapter 5.2.3) and informed user-server assignment (see Chapter 5.3.2) problems. Multiple network related metrics are computed for the ISPs network, including the path delays within the ISP network, maximum link utilization, and number of utilized backbone router hops. For the CDIs we compute the network statistics for their subset of the traffic as well as the load imposed on each CDI cluster and the network delay within the ISP topology for each cluster. By using an appropriate pricing model it is therefore possible for the CDIs to estimate their economic benefits.

## 7.2 Datasets

**Commercial CDI Dataset**: The CDI dataset covers a two-week period from 7th to 21st March 2011. All entries in the log we use relate to the Tier-1 ISP. This means that either the server or the end-user is using an IP address that belongs to the address space of the Tier-1 ISP. The CDI utilizes a highly distributed content delivery architecture, see Chapter 3.2, and operates a number of server clusters located inside the ISP and uses IPs in the IP address space of the ISP. The log contains detailed records of about 62 million sampled (uniformly at random) valid TCP connections between the CDI servers and end-users. For each reported connection, it contains the time it was recorded, the server IP address, the cluster the server belongs to, the anonymized client IP address, and various connection statistics such as bytes sent/received, duration, packet count and RTT. The CDI operates a number of services, utilizing the same infrastructure, such as dynamic and static web pages delivery, cloud acceleration, and video streaming.

**ISP Dataset**: The ISP dataset consists of two parts. First, detailed network information about the Tier-1 ISP, including the backbone topology, with interfaces and link annotations such as routing weights, as well as nominal bandwidth and delays. It also contains the full internal routing configuration which includes all subnets propagated inside the ISP either from internal routers or learned from peerings. The ISP operates more than 650 routers in about 500 locations (PoPs), and 30 peering points worldwide. We analyzed more than 5 million routing entries to derive a detailed ISP network view.

The second part of the ISP dataset is an anonymized packet-level trace of residential DSL connections. Our monitor, using Endace monitoring cards [44], observes the traffic of around 20,000 DSL lines to the Internet. Figure 7.1 shows a schematic of our monitoring setup. We capture HTTP and DNS traffic using the Bro IDS [134]. We observe 720 million DNS messages and more than 1 billion HTTP requests involving about 1.4 million unique hostnames. Analyzing the HTTP traffic in detail reveals that a large fraction it is due to a small number of CDIs, including the considered CDI, hyper-giants and One-Click Hosters [71, 98, 111] and that more than 65% of

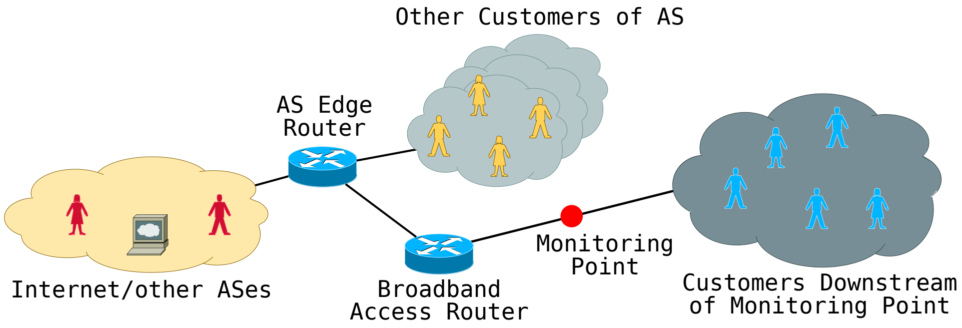


Figure 7.1: Schematic for data measurement setup.

the traffic volume is due to HTTP. Note that the second part of the ISP dataset is described in more detail in Chapter 4.1.

To derive the needed traffic matrices, on an origin-destination flow granularity, we compute from the DSL traces (on a 10-minute time bin granularity) the demands for the captured location in the ISP network. This demand is then scaled according to the load imposed by users of the CDI to the other locations in the ISP network. For CDIs without available connection logs, we first identify their infrastructure locations using the infrastructure aggregation approach as proposed by Poese et al. [137] and then scale the traffic demands according to the available CDI connection logs.

## 7.3 Collaboration Potential

We first describe our observations on the traffic and deployment of the large commercial CDI inside the Tier-1 ISP and analyze the potential benefits of CDI-ISP collaboration. In Figure 7.2, we plot the normalized traffic (in log scale) from CDI clusters over time. We classify the traffic into three categories: *a)* from CDI servers inside the ISP to end-users inside the ISP (ISP → ISP), *b)* from servers outside the ISP to end-users inside the ISP (outside → ISP), and *c)* from CDI servers inside the ISP to end-users outside the ISP (ISP → outside).

We observe the typical diurnal traffic pattern and a daily stability of the traffic pattern. Over the two week measurement period, 45.6% of the traffic belongs to the ISP → ISP category. 16.8% of the traffic belongs to the outside → ISP category. During peak hours, outside → ISP traffic can grow up to 40%. Finally, 37.6% of the traffic is served by inside clusters to outside end-users. Our first important observation is that a significant fraction of the CDI traffic is served from servers outside the ISP despite the presence of many servers inside the ISP that would be able to serve this traffic.

Figure 7.3 shows the re-allocation of traffic that would be possible using informed user-server assignment. Each full bar shows the fraction of traffic currently traversing

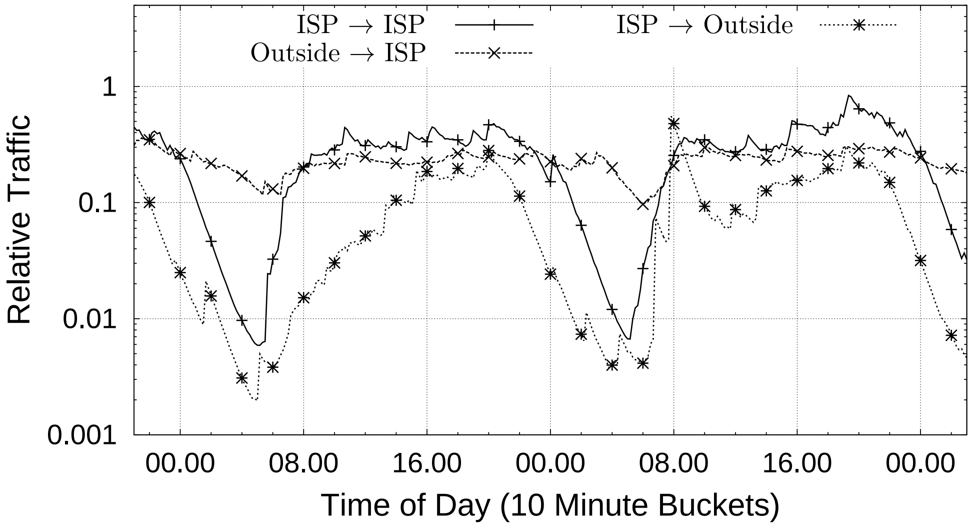


Figure 7.2: Activity of CDI in two days.

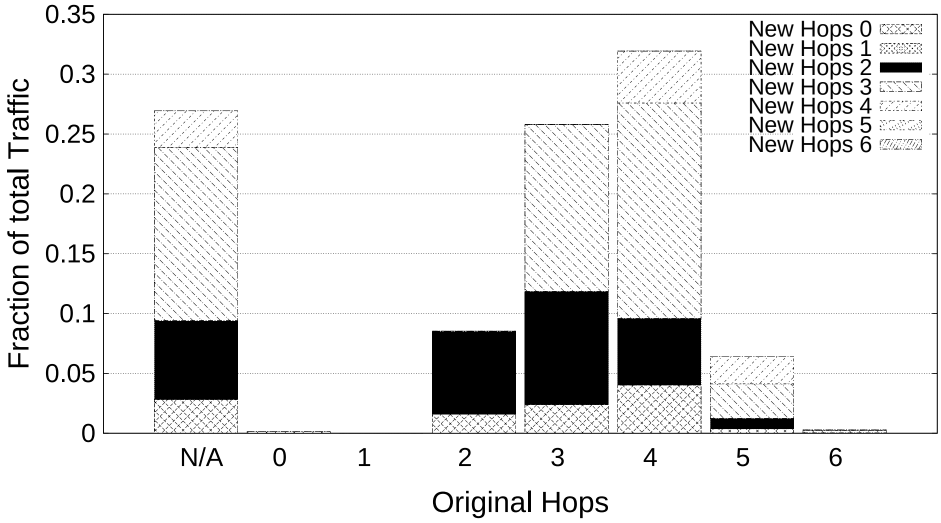


Figure 7.3: Potential hop reduction by using NetPaaS.

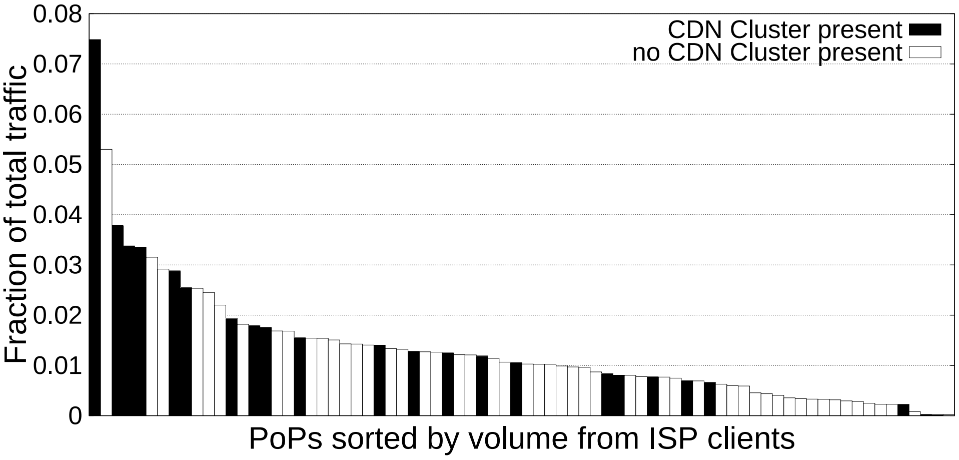


Figure 7.4: Traffic demand by ISP network location.

a given number of router hops within the ISP network. In this evaluation, we only consider the end-users inside the ISP. The bar labeled "N/A" is the traffic of the outside → ISP category. The different shaded regions in each bar correspond to the different router hop distances after re-allocation of the traffic. Almost half of the traffic currently experiencing 3 hops can be served from a closer-by server. Overall, a significant fraction of the traffic can be mapped to closer servers inside the ISP. Note that the tiny amount of traffic for router hop count 0 and 1 is due to the topology design of the ISP network: either the traffic stays within a PoP or it has to traverse at least two links to reach another PoP.

In Figure 7.4, we show the traffic demand towards the CDI generated by each PoP. We observe that some PoPs originate high demand while others have limited demand, if any. Manual inspection reveals that some of the PoPs with high demand cannot be served by a close-by CDI server, while other low demand PoPs have a cluster near by. Variations in the demand over time exhibit even more significant mismatches between demand and CDI locations. With such a time-varying demand and the timescales at which CDI deployments take place today, such mismatches should be expected.

We conclude that there are ample opportunities for CDIs to benefit from collaboration with ISPs to re-arrange or expand their footprint. Also, these observations support the use of NetPaaS to improve the operation of both the CDI and the ISP in light of the new CDI-ISP strategic alliances [11, 13, 14, 15].

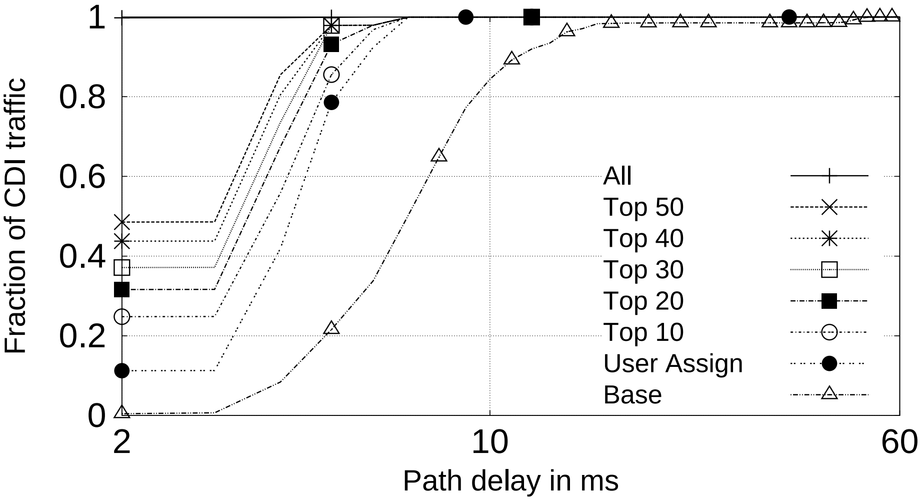
## 7.4 Improvements with NetPaaS

In this section we quantify the benefit of NetPaaS for the large commercial CDI inside the Tier-1 ISP. First we show the benefits of informed user-server assignment for the existing CDI infrastructure and continue with the additional benefit of in-network server allocation. In our evaluation we ensure that NetPaaS respects the available CDI server capacities and specifications in different locations. In the rest of the section, unless otherwise mentioned, we optimize the delay between end-user and CDI server [128]. Moreover, as we will show in our evaluation, by optimizing the delay between end-user and CDI server other traffic engineering goals are achieved.

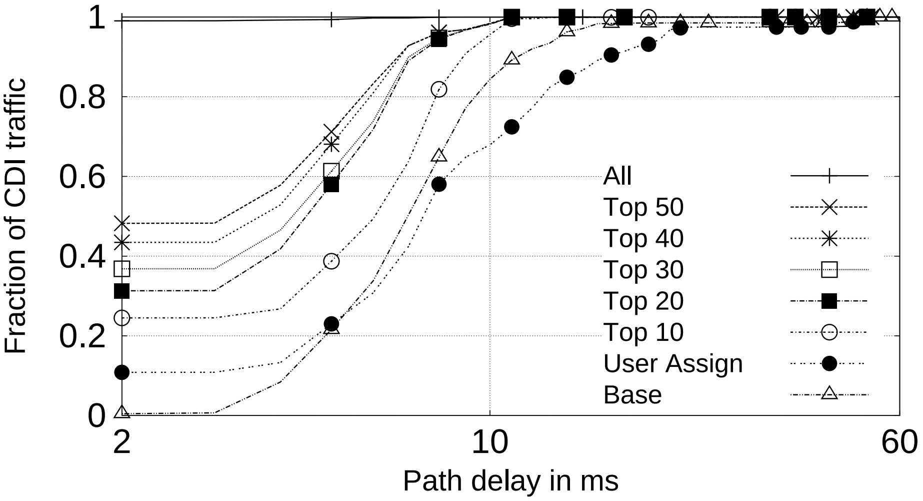
### 7.4.1 Informed End-User to Server Assignment

We first evaluate the benefits NetPaaS can offer when using informed user-server assignment only for the already deployed infrastructure of the large commercial CDI. In Figure 7.5 we show the current path delay between end-users and CDI servers, annotated as “Base”, and the resulting path delay after using informed user-server assignment, annotated as “User assign”. When optimizing the delay between end-users and CDI servers, see Figure 7.5a, the delay is reduced by 2–6 ms for most of the CDI traffic and another 12% of all traffic can be fetched from nearby CDI servers, a significant performance gain. To achieve similar gains CDIs have to rely on complicated routing tweaks [96]. When considering other optimization goals the gains are less prominent and in some cases even increases the delay. Figure 7.5b shows the current path delay between end-users and CDI servers when the optimization goal is to minimize the maximum link utilization. Again, informed user-server assignment enables around 12% of all traffic to be fetched from nearby CDI servers and slightly improves the delay of more than 20% of the traffic. However, around 70% of the traffic experience an modest increases in delay by up to 5 ms and the remaining 10% by up to 20 ms. Nevertheless, this behaviour is expected as the optimization goal completely ignores the path delay in favor of reducing the maximum link utilization. Therefore, an increased delay between end-users and CDI servers comes at no surprise.

When utilizing NetPaaS for informed user-server assignment the traffic traverses a shorter path within the network. This yields an overall traffic reduction in the network. In Figure 7.6 we plot the reductions in the overall traffic within the network, labeled “User-assign”. The results show a strong diurnal pattern that fit our activity observations in Figure 7.2 and indicate that informed user-server assignment enables major traffic savings when especially valuable, namely during peak hours. When optimizing for delay or maximum link utilization, the network wide traffic reduction can be as high as 7% during the peak hour, see Figure 7.6a. However, there is still room for improvement and by optimizing for a low number of utilized backbone router hops, the traffic savings can go up to more than 10%, see Figure 7.6b. This is a significant traffic volume that is on the scale of tens to hundreds of Terabytes per



(a) Optimizing the delay between end-user and CDI server.



(b) Optimizing the maximum link utilization.

Figure 7.5: Improvements in user to server delay.

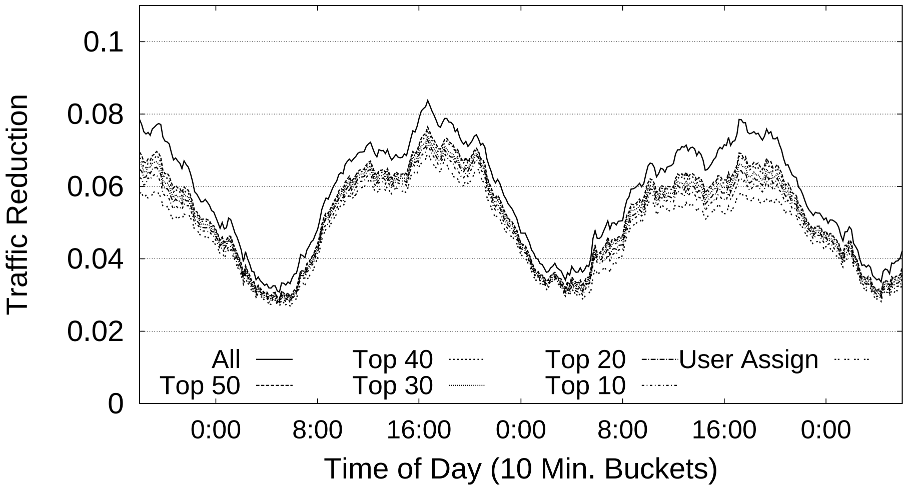
day in large ISPs [24, 55]. As a consequence, the most congested paths are circumvented, as the full server and path diversity is utilized [140]. Our evaluation shows that informed user-server assignment significantly improves CDI operation with the already deployed infrastructure and capacity. Moreover, the ISP does not need to change its routing, thus reducing the possibility of introducing oscillations [65].

In Figure 7.7 we plot the reduction in utilization for the most congested link at any point of time. We observe that during the peak time the utilization of the most congested link can be reduced by up to 65%. This is possible as traffic is better balanced and the link is utilized to serve mainly the local demand. Such a reduction in utilization can postpone link capacity upgrades. We observe very similar results for the different optimization goals, e.g., around 70% when optimizing for maximum link utilization. We suspect that both the high reduction and the small differences to other optimization goals can be explained by two facts: First, during peak hours nearly 40% of the CDI traffic is fetched from outside CDI servers and at the same time a similarly large amount of traffic is send from CDI servers inside the ISP to end-users outside the ISP. Second, the considered link is located in a city that accommodates the world's largest traffic exchange infrastructure and can be considered the most important Internet exchange in Central Europe. Therefore, utilizing the already available CDI server infrastructure inside the ISP enables either optimization goal to prevent most of the traffic from crossing the mentioned link.

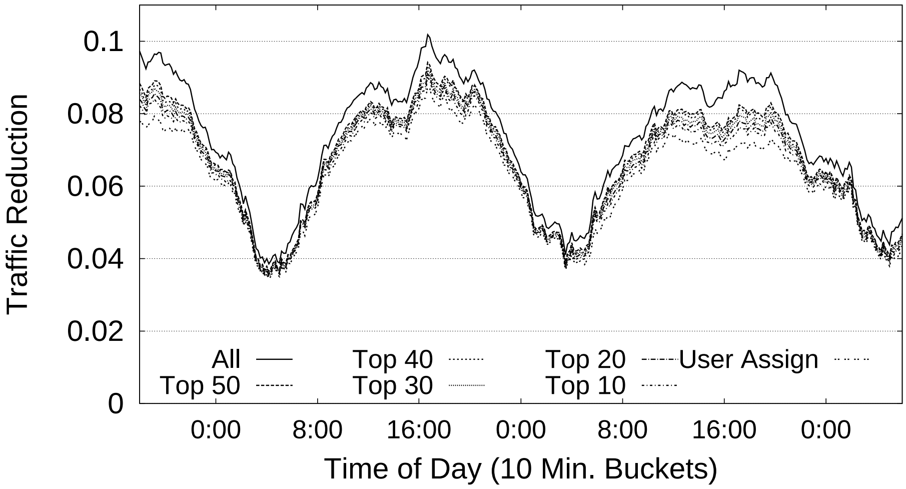
### 7.4.2 In-network Server Allocation

We next evaluate the benefits of NetPaaS when in-network server allocation is used in addition to informed user-server assignment. For short term CDI server deployments virtualized servers offer flexibility. For long term deployments, especially in light of the CDI-ISP alliances [11, 13, 14, 15], bare metal servers offer better performance. As our evaluation shows, the optimized placement of servers improves end-user performance as well as server and path diversity in the network, and enables ISPs to achieve traffic engineering goals, such as reducing the network wide traffic.

To estimate the locations for installing new servers, we use the local search heuristic to approximate the solution of CFL (see Chapter 6.2.3). Figure 7.8 shows the accuracy of in-network server allocation in terms of delay reduction when deploying 30 and 50 additional servers, labeled "Top 30" and "Top 50" respectively (similar observations are made for other numbers of servers). Notice that these 30 or 50 servers are not necessarily in the same PoP. It can be the case that more than one server is in the same PoP. For the optimal cases we pre-compute the best server locations based on the full knowledge of our 14-days dataset, while NetPaaS calculates the placement by utilizing past traffic demands and the current network activity during runtime. Our results show that NetPaaS achieves gains close to those of the optimal placement.



(a) Optimizing the delay between end-user and CDI server.



(b) Optimizing the number of backbone router hops.

Figure 7.6: Total traffic reduction within the ISP network.

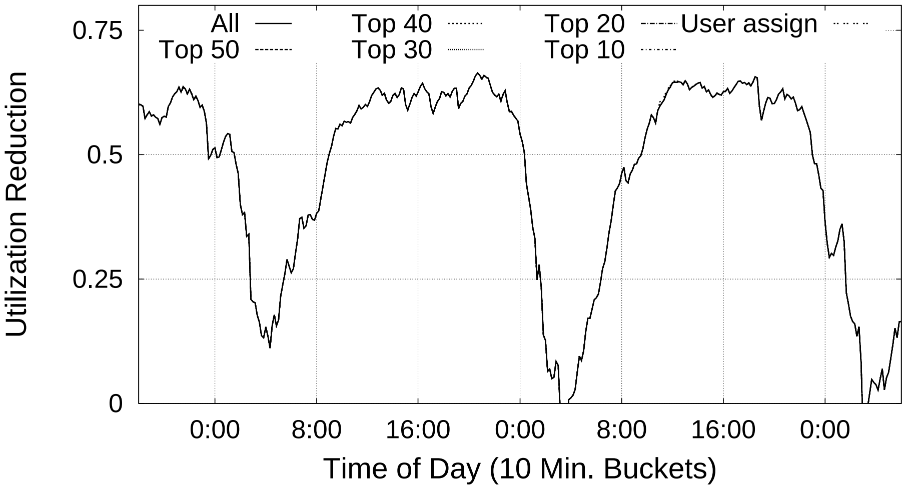


Figure 7.7: Maximum link utilization reduction.

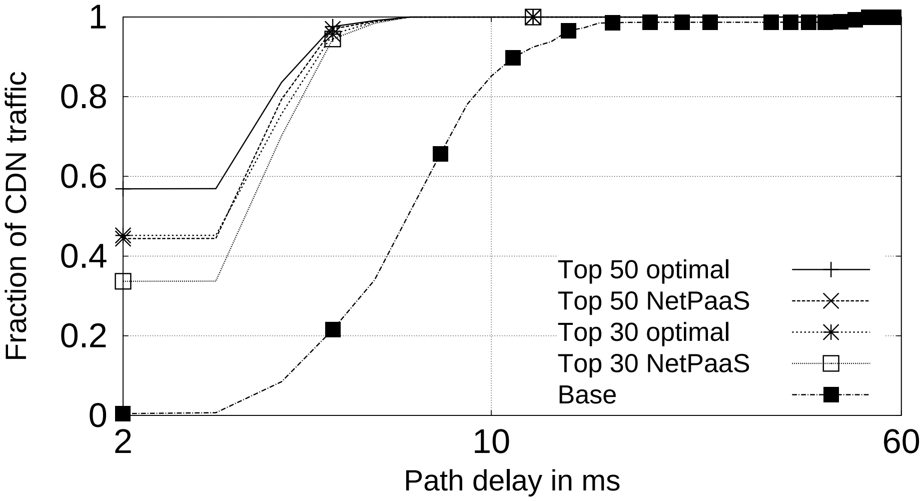
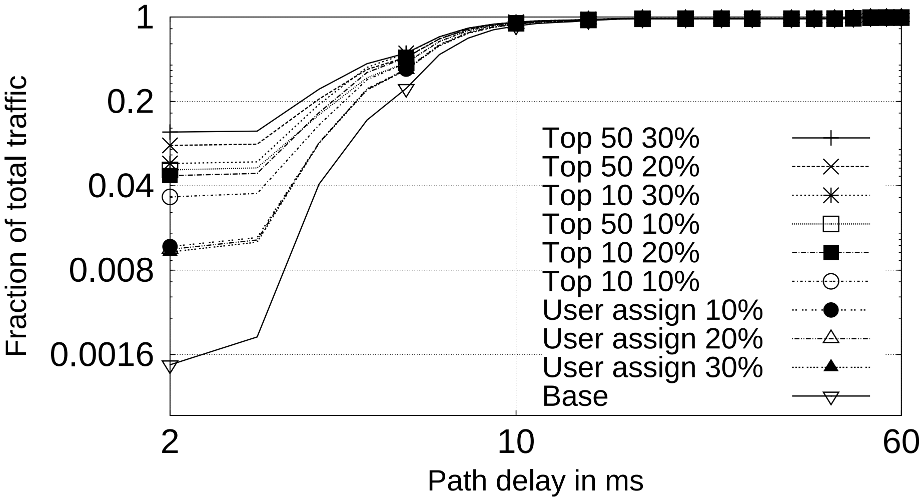


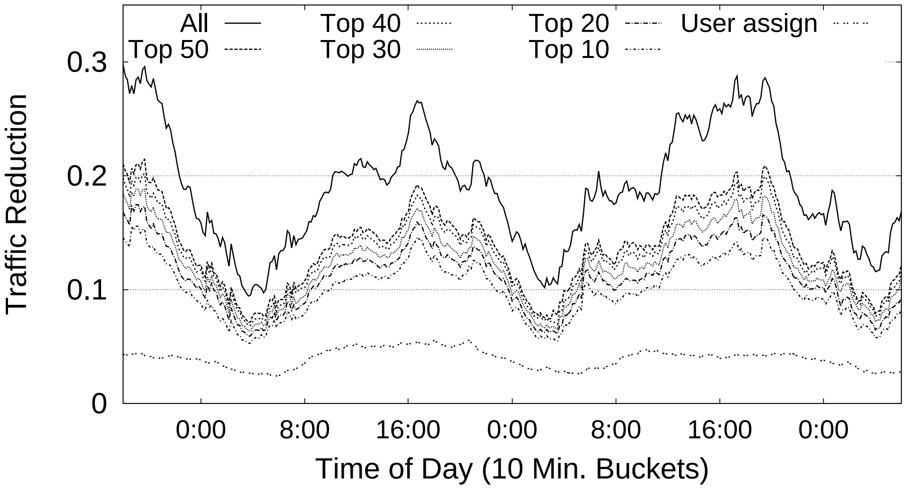
Figure 7.8: NetPaaS accuracy in selecting server location.

In Figure 7.5 we show the delay improvements of NetPaaS when less than 10% of the possible additional servers are utilized, thus we range the number of servers between 10 to 50 servers that are allocated in any of the about 500 locations within the ISP, labeled “Top 10” to “Top 50”. We also include a case where servers are allocated in all possible locations, labelled “All”. As expected, in this pathological case nearly all traffic can be served from the same PoP as the end-user. Yet, with only 10 additional servers around 25% of the CDI demand can be satisfied in the same PoP. With 50 additional servers it is possible to satisfy more than 48% of the CDI demand by a server located in the same PoP as the end-users. This achievement is independent of the chosen optimization goal and is expected, as using a CDI server in the same PoP is the best choice for each considered optimization goal. The difference of the optimization goals can be seen in the remaining traffic characteristics. Optimizing for delay between end-users and CDI servers yields to most significant improvements. Nearly all traffic can be fetched within 5 ms, more precise 85% with 10 additional servers and up to 97% with 50, see Figure 7.5a. When the optimization goals considers the maximum link utilization, see Figure 7.5b, nearly 40% of the traffic can be fetched within 5 ms with 10 additional servers while this amount goes up to 70% with 50 additional servers. With 20 to 50 additional servers around 98% of all traffic is in range of 11 ms and 95% within 18 ms with 10 additional servers. This shows that a relatively small number of servers can reduce the end-user to server delay significantly. It also shows the impact that the placement of a server plays in reducing the delay between end-user and content server. Note, that we report on the reduction of the backbone delay, the reduction of the end-to-end delay is expected to be even higher as the server is now located in the same network.

We next turn our attention to the possible traffic reduction in the network when NetPaaS is used. In Figure 7.6 we show the possible network wide traffic reduction with in-network server allocation when 10 to 50 servers can be allocated by the CDI. When optimizing the end-user delay, the traffic reduction especially during the peak hour ranges from 7% with 10 additional servers and reaches up to 7.5% when 50 additional servers can be utilized, see Figure 7.6a. The traffic savings increase up to 9.4% when optimizing for the number of utilized backbone router hops, but the overall shape remains the same, see Figure 7.6b. Again, this is a significant traffic volume that is on the scale of tens to hundreds of Terabytes per day in large ISPs. Note that the primary goal of NetPaaS in Figure 7.6a was to reduce the end-user to server delay, not network traffic. If all available locations (about 500) are utilized by the CDI, then the total traffic reduction during peak time is around 8% when optimizing for end-user delay and more than 10% when optimizing for backbone router hops. This shows that a small number of additional servers significantly reduces the total traffic inside the network. We also notice that our algorithm places servers in a way that the activity of the most congested link is not increased, see Figure 7.7. In our setting, further reduction of the utilization of the most congested link by adding more servers was not possible due to routing configuration.



(a) Reduction in user-server delay.



(b) Total traffic reductions (30% CDI traffic).

Figure 7.9: Joint service deployment with NetPaaS.

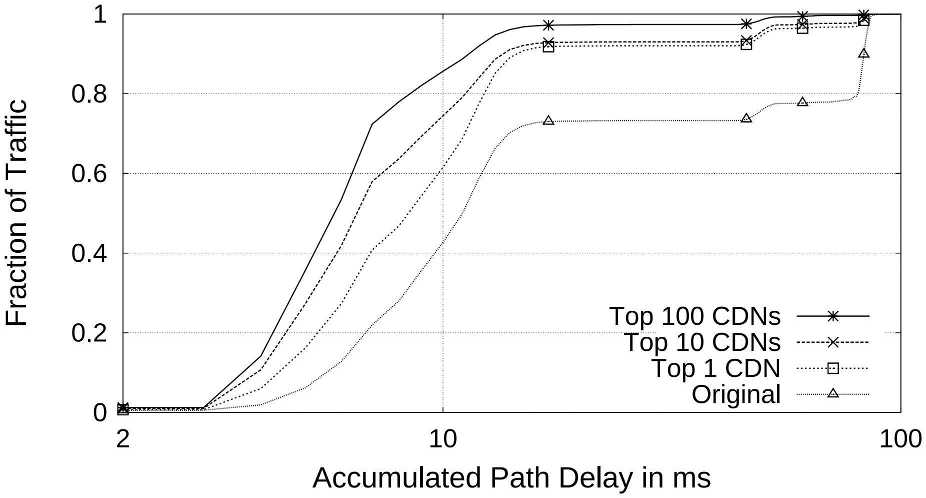
## 7.5 Joint Service Deployment with NetPaaS

We next consider the case of a CDI or an application that is launched within an ISP by exclusively utilizing NetPaaS. Examples include ISP-operated CDIs, licensed CDIs, or application-based CDIs. The latter is already happening with Google Global Cache [75] and with Netflix Open Connect in North America and North Europe [122]. Today, Netflix is responsible for around 30% of the total traffic in the peak hour in major US-based carriers [151]. We evaluate the performance of NetPaaS when such a service is launched and utilizes the system for a collaborative deployment of server resources. In Figure 7.9 we show the benefits of a joint CDI-ISP server deployment within the network. For our evaluation, we use the large commercial CDI, for which we know the sources of the demand and the server specifications and locations, and scale its traffic to reach 10%, 20%, or 30% of the total traffic of the ISP. As previously, with NetPaaS and using only informed user-server assignment, it is possible to satisfy a significant fraction of the total traffic from close-by servers, see Figure 7.9a. This can be even increased further when additional locations are available via in-network server allocation. Our results also show that while increasing the traffic demand for the CDI, NetPaaS manages to keep the delay between users and servers low, as well as to reduce the total network traffic.

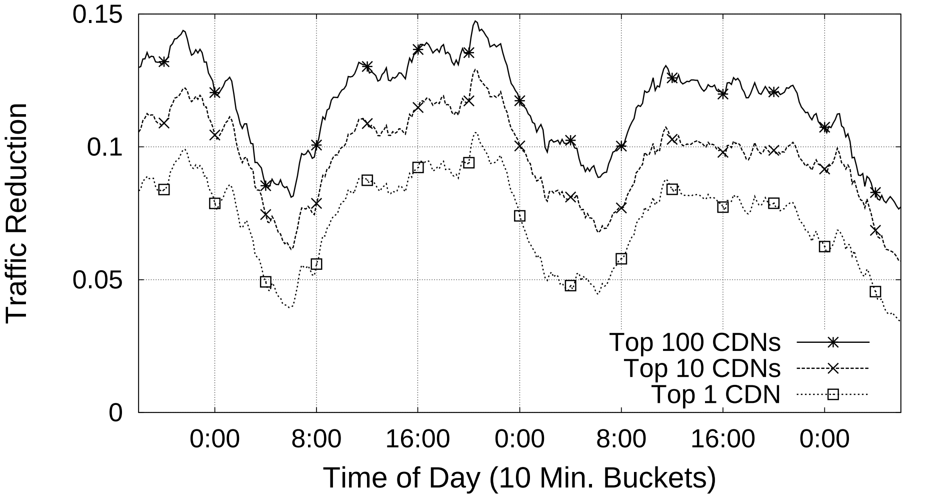
Figure 7.9b shows the total traffic reduction when the CDI traffic accounts for 30% of the total traffic. With informed user-server assignment only, NetPaaS is able to reduce the total traffic inside the network by up to 5%. When assigning additional servers, NetPaaS is able to reduce the total traffic from 15% with 10 servers to 20% with 50 servers and with servers in all PoPs to 30% traffic reduction is possible.

## 7.6 Collaboration with multiple CDIs

We also tested NetPaaS with multiple CDIs to evaluate the scalability of the system as well as the potential benefit of the system. For this, only informed user-server assignment was used as no information about the server requirements and the capacity of the other CDIs is available. We consider the top 1, 10, and 100 CDIs by traffic volume in the ISP. The largest CDI accounts for 19% of the total traffic, the top 10 CDIs are responsible for more than 40% and the top 100 CDIs for more than 67% respectively. Most of the large CDIs have deployed distributed infrastructure, located in a number of networks [140]. Figure 7.10 shows the improvements in user-server delay as well as the total traffic reduction achieved by NetPaaS. For the largest CDI most of the traffic can be served from close-by servers and as a result the total traffic can be reduced by up to 10%. When turning our attention to the top 10 and top 100 CDIs, we observe that NetPaaS is able to further increase the improvements, but with diminishing returns. With the top 10 CDIs the traffic is reduced by up to 13% and with the top 100 CDIs 15% respectively. We conclude that NetPaaS is able to achieve most of the benefits with the top 10 CDIs.



(a) Reduction in user-server delay.



(b) Total network traffic reduction.

Figure 7.10: Improvements with NetPaaS considering the Top 1, 10, and 100 CDIs.

## 7.7 Summary

In this Chapter, we evaluate NetPaaS, a system to orchestrate the on-demand allocation and assignment of resources in microdatacenters – additional server resources inside the network – by utilizing the ISPs view about the network. We quantify the possible benefits of CDI-ISP collaboration by analyzing operational traces from the largest commercial CDI and a European Tier-1 ISP. We consider multiple important network related metrics and quantify by how much NetPaaS can improve them. To this end, we also consider different optimization goals, such as minimizing the end-user to server delay, network wide traffic and maximum link utilization.

We start by analyzing the CDI-ISP collaboration potential. We find that during peak hours up to 40% of the traffic for end-users inside the ISP is fetched from servers outside the ISP. Around 37.6% traffic from servers inside the ISP are served to outside end-users. In addition, we find a significant mismatch between end-user demand and CDI server locations inside the ISP.

Utilizing informed user-server assignment and optimizing for delay, NetPaaS enables 12% of the traffic to be fetched from nearby servers while reducing the delay by 2-6 ms. In addition, the network wide traffic is reduced by up to up to 7.5%. This increases up to 10% when optimizing for network traffic. Furthermore, NetPaaS is able to reduce utilization of the most congested link by up to 65%, which increases up to 70% when optimizing for link utilization. When utilizing in-network server allocation with 50 additional servers, NetPaaS is able to satisfy 48% of the CDI demand from the same PoP and up to 97% of all CDI traffic can be fetched within 5 ms delay to the end-user (around 70% when optimizing for maximum link utilization). Moreover, NetPaaS reduces the network wide traffic by up to 8% during peak hours (10% when optimizing for traffic volume). We note that in no case the most utilized link was further congested. Our results highlight the importance of server placement and that already few additional servers can significantly reduce the end-users delay and the network wide traffic. Our evaluation of joint service deployment utilizing NetPaaS shows that a significant amount of traffic can be served from nearby locations, even with increasing demands of up to 30% of the total traffic. Last but not least, we investigate the scalability of NetPaaS and show its ability to facilitate collaboration with multiple CDIs including increased benefits for all parties.

We conclude that NetPaaS enables joint optimization of CDI operation and deployment. It enables a collaborative approach to better utilize existing CDI infrastructures, allows dynamic allocation of additional resources inside the ISPs network and at the same time improves important network metrics such as end-user to server delay, maximum link utilization or network wide traffic.

# 8
# Conclusion

Content Delivery Infrastructures are responsible for a significant fraction of todays Internet traffic [49,71,98,137]. To improve their delivery performance, increase their infrastructure footprint, and to reduce capital investment and operational costs, CDIs recently started to look into novel content delivery architectures, such as Hybrid CDIs and Federated CDIs [3, 80, 124, 143]. However, most of these infrastructures are entangled with the very infrastructures that provide network connectivity to end-users [67, 102]. Despite putting tremendous effort into discovering end-to-end characteristics to predict performance [37,96,128], none of the current and emerging CDI architectures has so far considered *collaboration* to obtain them.

In this thesis, we assess the impact of collaboration between CDIs and ISPs on content delivery. Based on our findings, we argue that CDI-ISP collaboration is the next step in the natural evolution of content delivery infrastructures. This is challenging since today there is no system that facilitates CDI-ISP collaboration. To this end, we propose a novel system design, called NetPaaS, that enables CDI-ISP collaboration in near real-time without revealing any sensitive operational data.

## 8.1 Summary and Implications

We start our assessment of collaboration by discussing the challenges CDIs and ISPs face in content delivery today. We then provide a systematic evaluation of the design and operating space of content delivery architectures. We find that the content delivery landscape is in a constant flux to further improve its delivery performance and increase its network footprint, while at the same time tries to reduce the capital investment and operational costs for its content delivery infrastructure. However, the lack of agility in server deployment as well as limited knowledge about the state of the underlying network are still open problems in any proposed CDI architecture. Therefore, we believe that collaboration between CDIs and ISP is the key for an architecture independent solution.

To this end, we ask how much benefit such a CDI-ISP collaboration can potentially offer. We investigate possible benefits by analyzing operational traces from an European Tier-1 ISP. We find that collaboration during the assignment of end-users to CDI servers increases the traffic localization potential two-fold and highly improves the end-user performance. Furthermore, already existing path diversity is significant and enables new mechanisms for managing traffic flows inside the ISPs network. Thus, our findings support our view that CDI-ISP collaboration can greatly improve todays content delivery.

To facilitate CDI-ISP collaboration, we propose two key enablers that allow the coordination of CDIs and ISPs. The first enabler, *in-network server allocation*, coordinates the placement of servers within a network between CDIs and ISPs. It provides an additional degree of freedom to the CDI to scale-up or shrink the footprint on demand and enables agile allocation of additional resources close to the end-users. With recent advantages in virtualization technology and the comprehensive deployment of general purpose hardware inside the network, a more agile cloud style allocation of content delivery infrastructure is now possible. In contrast, the traditional way of deploying content delivery infrastructure was a tedious, time consuming and inflexible process associated with high capital investment and operational costs. The second enabler, *informed user-server assignment*, allows CDIs to receive recommendations from an ISP, i.e., a server ranking based on performance criteria mutually agreed upon. The recommendation allows the ISP to take possible network bottlenecks into account and at the same time enables it to influence how the traffic flows through its network thus reducing the network traffic volatility, something CDIs can not achieve on their own. Moreover, the precise knowledge about the end-users location and the current network conditions allows the ISP to effectively select the best possible candidate server for each individual end-user request. Therefore, both enablers increases customer satisfaction through performance improvements and enable simplified network provisioning and traffic engineering leading to a win-win situation.

We implement these principles in NetPaaS (Network Platform as a Service), a novel system that orchestrates the on-demand deployment of services by utilizing the ISPs view about the network and cloud-style resources inside the ISPs network. Using up-to-date network information, NetPaaS offers unprecedented flexibility and performance improvements through agile infrastructure deployment and informed end-user to server assignment without revealing sensitive operational data. Based on our design and the proposed algorithms, we argue that near real-time collaboration between CDIs and ISPs is within reach of todays technology and can be seen as the next step in the evolution of scalable and efficient content delivery architectures.

To quantify the potential of CDI-ISP collaboration with NetPaaS, we perform a first-of-its-kind evaluation based on operational traces from the largest commercial CDI and an European Tier-1 ISP. Our findings reveal how important accurate and up-to-date information about the end-user locations and network conditions are to the CDI, especially when allocating additional server resources inside the ISPs network.

First, we find a significant mismatch between end-user demand and CDI locations inside the ISP, especially during peak hours. Our evaluation of informed user-server assignment shows that by utilizing the ISPs view on the network, nearly all of the CDIs traffic could be served from clusters within the ISP. Thus, CDI-ISP collaboration offers major performance improvements and enables the CDI to better utilize existing server resources inside the ISP. At the same time, enables the ISP to better manage traffic flows inside its network and to achieve traffic engineering goals.

Second, we find that the dynamic allocation of a small number of additional servers with in-network server allocation alleviates the large mismatch of end-user demand and CDI server locations. Leveraging up-to-date network information from the ISP when allocating new servers allows the CDI to better cope with the increasing and highly volatile demand for content. Moreover, in-network server allocation greatly improves the end-users performance and reduces the network wide traffic by a significant fraction. Therefore, it improves the ISPs ability to manage and engineer a large fraction of its traffic and leads to a win-win situation for all parties.

Third, our evaluation shows the scalability of NetPaaS. For this, we anticipate the launch of a traffic heavy service inside the ISP that exclusively relies on NetPaaS. Our results show that, even with demands of up to one third of the ISPs traffic, nearly all demand can be served from nearby locations when using only a small number of dynamically allocated servers. Furthermore, we show that NetPaaS can collaborate with multiple independent CDIs at the same time and that collaboration with a small number of large CDIs already offers most of the possible benefits.

We conclude that NetPaaS provides a novel mechanism for agile server deployment and informed end-user to server assignment based on up-to-date network information. NetPaaS offers unprecedented flexibility to deploy CDI server on demand. It improves the scalability, efficiency and performance of content delivery infrastructures and at the same time enables ISPs to achieve traffic engineering goals. Utilizing cloud style resources inside the network enables CDIs to greatly improving the performance of content delivery for the end-user and at the same time reduces the needed capital investment and operational costs.

## 8.2 Future Work

We see several different directions for future research arising from the results of this thesis. So far, the evaluation of NetPaaS focused on a specific and well known content delivery architecture, namely the highly distributed one. However, recently emerging architectural trends like Hybrid or Meta Content Delivery Infrastructures, are becoming more and more common, and thus open a new area of research for collaborative approaches like NetPaaS. Furthermore, our study concentrates on Web content delivered over HTTP to evaluate the benefits of collaboration between CDIs and ISPs. However, streaming of video and audio is becoming more and more prevalent in the current Internet traffic mix. Therefore, we believe that investigating the specific properties and requirements of streaming applications, even when done on top of HTTP, is an important next step to further improve the benefits and efficiency of collaboration in content delivery.

**Hybrid CDIs** In Hybrid CDIs end-user contribute available resources, such as storage and bandwidth, in addition to the available server resources offered by the CDI. However, ISPs offer many different technologies for Internet access, e.g., xDSL, Cable and dial-up for fixed lines or 3G and LTE for wireless. Therefore, important network metrics for content delivery, such as delay, bandwidth, and also for how long a given resource is available vary a lot more. Furthermore, the content may be only partially or even no longer available at end-users. Thus, to enable NetPaaS to efficiently include end-user provided resources the next necessary steps include to review the system design and used protocols to support potentially up to hundreds of thousands of end-users in a scalable fashion.

**Meta-CDIs** Meta-CDIs are basically brokers for content delivery resources. These brokers collect performance metrics from a large number of end-users to determine the best CDI for individual end-users. They can be seen as an indirection layer between the end-user and the utilized CDIs. However, a Meta-CDI faces at least the same challenges just as any other CDI. Due to the additional indirection and the possibly not timely and eventually inaccurate end-user supplied measurements, these challenges are most probably exacerbated. Therefore, more research into the mechanics and selection criteria are important for collaborative approaches like NetPaaS to support Meta-CDIs.

**Streaming Applications** Video and audio streaming services, such as Netflix or Spotify, have seen tremendous growth, both in number of users and in traffic volumes, over the last years. The utilized streaming protocols are designed to reduce and mitigate the impact of delay and network bottlenecks to enable fluent data streaming for a smooth user experience. However, our system design of CDI-ISP collaboration has so far neglected any additional information that could be provided by the application layer, such as the streaming bitrate or the expected runtime (i.e., video length). Therefore, additional research into application layer provided information seems the next logical step to further improve the benefits and performance of collaborative approaches like NetPaaS.

# Acknowledgments

First and foremost, I am very grateful to my advisor and mentor Anja Feldmann for her patient guidance, encouragement and useful critiques of my work. Working with Anja has always been a pleasure to me.

I owe special thanks to my collaborators and friends Georgios Smaragdakis, Ingmar Poese and Steve Uhlig. I very much enjoyed all the time we spent together discussing and fighting over our ideas and solutions, even at 3 am.

My sincerest thanks go to Vinnay Aggarwal who first introduced me to the topic of CDI-ISP collaboration and since has become a dear friend.

My gratitude also goes to Bruce Maggs, who readily shared his tremendous knowledge of content delivery.

I want to thank Oliver Hohlfeld for his time proof reading and giving feedback to my thesis and all the nice and worthwhile discussions we had; Carlo and Arne for providing me with ample reasons to procrastinate; Bernhard Ager for the support and help in the early days of my Ph. D.

Special thanks go to our secretaries Britta, Birgitt and lately Nadine and our IT-Crowd Rainer, Bernd and Sandra for being so helpful and responsive even in cases of self-inflicted mayhem or yet another broken computer; and everyone of FG INET, I really enjoyed being a part of this awesome group.

Additionally, I want to thank the people at T-Labs for our nice and fruitful collaboration, especially Michael Düser and Andreas Gladisch for providing the ISP side of things.

Last but not least I want to thank my family for always supporting and encouraging me, especially my wife Verena and my parents Rosi and Hermann.

# List of Figures

# List of Tables

# Bibliography

[1] Abrams, M., Standridge, C. R., Abdulla, G., Fox, E. A., and Williams, S. Removal policies in network caches for world-wide web documents. In *ACM SIGCOMM* (1996).

[2] Adhikari, V., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M., and Zhang, Z.-L. Unreeling netflix: Understanding and improving multi-cdn movie delivery. In *IEEE INFOCOM* (2012).

[3] Aditya, P., Zhao, M., Lin, Y., Haeberlen, A., Druschel, P., Maggs, B., and Wishon, B. Reliable Client Accounting for Hybrid Content-Distribution Networks. In *ACM NSDI* (2012).

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

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

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

[7] Ager, B., Mühlbauer, W., Smaragdakis, G., and Uhlig, S. Web Content Cartography. In *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., Bender, S., Feldmann, A., and Wichmann, A. Methodology for Estimating Network Distances of Gnutella Neighbors. In *GI Jahrestagung - Informatik* (2004).

[10] Aggarwal, V., Feldmann, A., and Scheideler, C. Can ISPs and P2P systems co-operate for improved performance? *ACM SIGCOMM Computer Communication Review 37*, 3 (2007).

[11] Akamai. Akamai and AT&T Forge Global Strategic Alliance to Provide Content Delivery Network Solutions. http://www.akamai.com/html/about/press/releases/2012/press_120612.html.

[12] Akamai. Akamai Facts & Figures. http://www.akamai.com/html/about/facts_figures.html.

[13] Akamai. KT and Akamai Expand Strategic Partnership. http://www.akamai.com/html/about/press/releases/2013/press_032713.html.

[14] Akamai. Orange and Akamai form Content Delivery Strategic Alliance. http://www.akamai.com/html/about/press/releases/2012/press_112012_1.html.

[15] Akamai. Swisscom and Akamai Enter Into a Strategic Partnership. http://www.akamai.com/html/about/press/releases/2013/press_031413.html.

[16] Akella, A., Seshan, S., and Shaikh, A. An Empirical Evaluation of Wide-Area Internet Bottlenecks. In *ACM Internet Measurement Conference* (2003).

[17] Alicherry, M., and Lakshman, T. Network aware resource allocation in distributed clouds. In *IEEE INFOCOM* (2012).

[18] Alimi, R., Penno, R., and Yang, Y. ALTO Protocol. Internet-Draft draft-ietf-alto-protocol-16.txt, IETF Secretariat, May 2013.

[19] AMAZON Web Services. http://aws.amazon.com.

[20] Andersen, D., Balakrishnan, H., Kaashoek, M., and Morris, R. Resilient Overlay Networks. In *ACM SOSP* (2001).

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

[22] 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. Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[23] Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit, V. Local Search Heuristics for $k$-Median and Facility Location Problems. *SIAM J. on Computing* (2004).

[24] AT&T. AT&T Company Information. http://www.att.com/gen/investor-relations?pid=5711, 2012.

[25] Awerbuch, B., and Leighton, F. Multicommodity Flows: A Survey of Recent Research. In *ISAAC* (1993).

[26] Axelrod, M. The Value of Content Distribution Networks. African Network Operators' Group, May 2008.

[27] Azar, Y., Naor, J., and Rom, R. The competitiveness of on-line assignments. *J. Algorithms 18*, 2 (1995).

[28] Balamash, A., and Krunz, M. An overview of web caching replacement algorithms. *IEEE Communications Surveys Tutorials 6*, 2 (2004).

[29] Ballani, H., Costa, P., Karagiannis, T., and Rowstron, A. Towards Predictable Datacenter Networks. In *ACM SIGCOMM* (2011).

[30] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. Xen and the art of virtualization. *ACM SIGOPS Operating System Review 37*, 5 (2003).

[31] Berners-Lee, T. The Original HTTP as defined in 1991. `http://www.w3.org/Protocols/HTTP/AsImplemented.html`, 1991.

[32] Berners-Lee, T., Fielding, R., and Frystyk, H. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945, RFC Editor, May 1996.

[33] Bill Gates. Content is King. Microsoft Essay, Mar 1996.

[34] Bindal, R., Cao, P., Chan, W., Medved, J., Suwala, G., Bates, T., and Zhang, A. Improving Traffic Locality in BitTorrent via Biased Neighbor Selection. In *IEEE ICDCS* (2006).

[35] Bradford, R., Kotsovinos, E., Feldmann, A., and Schiöberg, H. Live Wide-Area Migration of Virtual Machines Including Local Persistent State. In *VEE* (2007).

[36] Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., and Shenker, S. Ethane: Taking Control of the Enterprise. In *ACM SIGCOMM* (2007).

[37] Cedexis. Cedexis Free Country Reports. `http://www.cedexis.com/country-reports/`.

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

[39] Charikar, M., Guha, S., Shmoys, D., and Tardos, E. A Constant Factor Approximation Algorithm for the k-median Problem. In *ACM STOC* (1999).

[40] Chiaraviglio, L., Mellia, M., and Neri, F. Minimizing isp network energy cost: formulation and solutions. *IEEE/ACM Trans. Networking 20*, 2 (Apr 2012).

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

[42] Church, K., Greenberg, A., and Hamilton, J. On Delivering Embarrasingly Distributed Cloud Services. In *HotNets* (2008).

[43] CISCO Global Visual Networking and Cloud Index. Forecast and Methodology, 2016-2017. `http://www.cisco.com`.

[44] Cleary, J., Donnelly, S., Graham, I., McGregor, A., and Pearson, M. Design Principles for Accurate Passive Measurement. In *Passive and Active Measurement Conference* (2000).

[45] Coffman, J. E., Garey, M., and Johnson, D. Approximation Algorithms for Bin Packing: A Survey. In *Approximation algorithms for NP-hard problems* (1997).

[46] Cohen, B. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems* (2003).

[47] Contavalli, C., van der Gaast, W., Leach, S., and Lewis, E. Client subnet in DNS requests. Internet-Draft draft-vandergaast-edns-client-subnet-01.txt, IETF Secretariat, Apr 2012.

[48] Contavalli, C., van der Gaast, W., Leach, S., and Rodden, D. Client IP Information in DNS Requests. Internet-Draft draft-vandergaast-edns-client-ip-01.txt, IETF Secretariat, May 2010.

[49] Craig Labovitz. Latest Backbone Router Data: Massive Ongoing Changes in Content Distribution. `http://conferences.infotoday.com/documents/172/2013CDNSummit-B102A.pdf`, May 2013.

[50] Cronin, E., Jamin, S., Jin, C., Kurc, A., Raz, D., and Shavitt, Y. Constraint Mirror Placement on the Internet. *JSAC* (2002).

[51] Czumaj, A., Riley, C., and Scheideler, C. Perfectly Balanced Allocation. In *RANDOM-APPROX* (2003).

[52] Davison, B. Brian d. davison's web caching bibliography. `http://www.web-caching.com/biblio.html`, 2006.

[53] des Ligneris, B. Virtualization of linux based computers: the linux-vserver project. In *International Symposium on High Performance Computing Systems and Applications* (2005), IEEE, pp. 340–346.

[54] Deutsche Telekom. T-Systems to offer customers VMware vCloud Datacenter Services. `http://www.telekom.com/media/enterprise-solutions/129772`.

[55] Deutsche Telekom. Deutsche Telekom ICSS. `http://ghs-internet.telekom.de/dtag/cms/content/ICSS/en/1222498`, 2012.

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

[57] Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., and Weihl, B. Globally distributed content delivery. *IEEE Internet Computing* (2002).

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

[59] 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 *ACM SIGCOMM* (2011).

[60] Drago, I., Mellia, M., M Munafo, M., Sperotto, A., Sadre, R., and Pras, A. Inside dropbox: understanding personal cloud storage services. In *ACM Internet Measurement Conference* (2012), ACM Press.

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

[62] ETSI. European Telecommunications Standards Institute. `http://www.etsi.org/`, 2013.

[63] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, RFC Editor, Jun 1999.

[64] Fortz, B., and Thorup, M. Internet Traffic Engineering by Optimizing OSPF Weights. In *IEEE INFOCOM* (2000).

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

[66] Frank, B., Poese, I., Lin, Y., Smaragdakis, G., Feldmann, A., Maggs, B. M., Rake, J., Uhlig, S., and Weber, R. Pushing CDN-ISP Collaboration to the Limit. *ACM SIGCOMM Computer Communication Review 43*, 3 (2013).

[67] Frank, B., Poese, I., Smaragdakis, G., Feldmann, A., Maggs, B. M., Uhlig, S., Aggarwal, V., and Schneider, F. Collaboration Opportunities for Content Delivery and Network Infrastructures. *ACM SIGCOMM ebook on Recent Advances in Networking* (2013).

[68] Frank, B., Poese, I., Smaragdakis, G., Uhlig, S., and Feldmann, A. Content-aware Traffic Engineering. In *ACM SIGMETRICS* (2012).

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

[70] Freedman, M. J. Experiences with CoralCDN: A Five-Year Operational View. In *ACM NSDI* (2010).

[71] Gerber, A., and Doverspike, R. Traffic Types and Growth in Backbone Networks. In *Optical Fiber Communication/National Fiber Optic Engineers Conference* (2011).

[72] Gibb, G., Zeng, H., and McKeown, N. Outsourcing network functionality. In *HotNets* (2012).

[73] Gnutella v0.6 RFC. `http://www.the-gdf.org/`.

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

[75] Google. GoogleCache. `http://ggcadmin.google.com/ggc`.

[76] Google Datacenters. `http://www.google.com/about/datacenters/`.

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

[78] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., and Shenker, S. NOX: Towards an Operating System for Networks. *ACM SIGCOMM Computer Communication Review 38*, 3 (2008).

[79] Halabi, S. *Internet Routing Architectures.* Cisco Press, 2000.

[80] Huang, C., Li, J., Wang, A., and Ross, K. W. Understanding Hybrid CDN-P2P: Why Limelight Needs its Own Red Swoosh. In *NOSSDAV* (2008).

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

[82] Jain, K., and Vazirani, V. Primal-Dual Approximation Algorithms for Metric Facility Location and $k$-Median Problems. In *FOCS* (1999).

[83] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Holzle, U., Stuart, S., and Vahdat, A. Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure. In *ACM SIGCOMM* (2013).

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

[85] Kamp, P.-H., and Watson, R. N. Jails: Confining the omnipotent root. In *System Administration and Network Engineering* (2000), vol. 43, p. 116.

[86] Karagiannis, T., Rodriguez, P., and Papagiannaki, K. Should ISPs fear Peer-Assisted Content Distribution? In *ACM Internet Measurement Conference* (2005).

[87] Keralapura, R., Taft, N., Chuah, C., and Iannaccone, G. Can ISPs Take the Heat from Overlay Networks? In *HotNets* (2004).

[88] Khachiyan, L. A Polynomial Time Algorithm for Linear Programming. *Dokl. Akad. Nauk SSSR* (1979).

[89] Khare, R., and Lawrence, S. Upgrading to TLS Within HTTP/1.1. RFC 2817, RFC Editor, May 2000.

[90] Kleinberg, J., and Tardos, E. *Algorithm Design.* Addison-Wesley, 2005.

[91] 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).

[92] Korupolu, M., Plaxton, C., and Rajaraman, R. Analysis of a Local Search Heuristic for Facility Location Problems. *J. Algorithms 37* (2000).

[93] Korupolu, M., Singh, A., and Bamba, B. Coupled Placement in Modern Data Centers. In *IPDPS* (2009).

[94] Kotronis, V., Dimitropoulos, X., and Ager, B. Outsourcing the routing control logic: better internet routing based on sdn principles. In *HotNets* (2012).

[95] Krishnan, P., Raz, D., and Shavitt, Y. The Cache Location Problem. *IEEE/ACM Trans. Networking 8*, 5 (2000).

[96] 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 *ACM Internet Measurement Conference* (2009).

[97] Krishnan, S. S., and Sitaraman, R. K. Video Stream Quality Impacts Viewer Behavior: Inferring Causality using Quasi-Experimental Designs. In *ACM Internet Measurement Conference* (2012).

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

[99] Lageman, M. Solaris Containers—What They Are and How to Use Them. *Sun BluePrints OnLine* (2005), 819–2679.

[100] Laoutaris, N., Rodriguez, P., and Massoulie, L. ECHOS: Edge Capacity Hosting Overlays of Nano Data Centers. *ACM SIGCOMM Computer Communication Review 38*, 1 (2008).

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

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

[103] Lenstra, J., Shmoys, D., and Tardos, E. Approximation Algorithms for Scheduling Unrelated Parallel Machines. *Math. Program.* (1990).

[104] Levi, R., Shmoys, D., and Swamy, C. LP-based Approximation Algorithms for Capacitated Facility Location. In *SODA* (2004).

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

[106] Limelight Networks. `http://www.limelight.com/technology/`.

[107] Liu, B., Bi, J., and Yang, X. Faas: filtering ip spoofing traffic as a service. *ACM SIGCOMM Computer Communication Review 42*, 4 (2012).

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

[109] Liu, X., Dobrian, F., Milner, H., Jiang, J., Sekar, V., Stoica, I., and Zhang, H. A Case for a Coordinated Internet-Scale Video Control Plane. In *ACM SIGCOMM* (2012).

[110] Madhyastha, H., McCullough, J. C., Porter, G., Kapoor, R., Savage, S., Snoeren, A. C., and Vahdat, A. scc: Cluster Storage Provisioning Informed by Application Characteristics and SLAs. In *FAST* (2012).

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

[112] 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 ATC* (2002).

[113] Marocco, E., and Gurbani, V. Application-Layer Traffic Optimization. `http://datatracker.ietf.org/wg/alto/charter/`, 2008.

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

[115] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review 38*, 2 (2008).

[116] Mealling, M., and Denenberg, R. Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations. RFC 3305, RFC Editor, Aug 2002.

[117] Microsoft Azure. http://www.windowsazure.com.

[118] Mockapetris, P. Domain names - implementation and specification. RFC 1035, RFC Editor, Nov 1987.

[119] Morrison, D. R. Practical Algorithm to Retrieve Information Coded in Alphanumeric. *J. of the ACM* (1968).

[120] Moy, J. OSPF Version 2. RFC 2328, RFC Editor, Apr 1998.

[121] Nakao, A., Peterson, L., and Bavier, A. A Routing Underlay for Overlay Networks. In *ACM SIGCOMM* (2003).

[122] Netflix. Netflix Open Connect. https://signup.netflix.com/openconnect.

[123] NFV. Network Functions Virtualisation. SDN and OpenFlow World Congress, Oct 2012.

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

[125] Nottingham, M. AHypertext Transfer Protocol Bis. http://datatracker.ietf.org/wg/httpbis/charter/, 2007.

[126] Nottingham, M., and Fielding, R. Additional HTTP Status Codes. RFC 6585, RFC Editor, Apr 2012.

[127] Nottingham, M., and Mogul, J. HTTP Header Field Registrations. RFC 4229, RFC Editor, Dec 2005.

[128] 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).

[129] Olteanu, V. A., and Raiciu, C. Efficiently migrating stateful middleboxes. In *ACM SIGCOMM* (2012).

[130] Oracle. Virtualbox user manual, version 4.2.16. www.virtualbox.org, 2013.

[131] Oran, D. OSI IS-IS Intra-domain Routing Protocol. RFC 1142, RFC Editor, Feb 1990.

[132] Otto, J. S., Sánchez, M. A., Rula, J. P., and Bustamante, F. E. Content Delivery and the Natural Evolution of DNS - Remote DNS Trends, Performance Issues and Alternative Solutions. In *ACM Internet Measurement Conference* (2012).

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

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

[135] Peterson, L. Zen and the art of network architecture. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (2013), ACM SIGCOMM, pp. 1–2.

[136] Podlipnig, S., and Böszörmenyi, L. A survey of web cache replacement strategies. *ACM Comput. Surv. 35*, 4 (Dec 2003).

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

[138] Poese, I., Frank, B., Ager, B., Smaragdakis, G., Uhlig, S., and Feldmann, A. Improving Content Delivery with PaDIS. *IEEE Internet Computing 16*, 3 (2012).

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

[140] Poese, I., Frank, B., Smaragdakis, G., Uhlig, S., Feldmann, A., and Maggs, B. M. Enabling Content-aware Traffic Engineering. *ACM SIGCOMM Computer Communication Review 42*, 5 (2012).

[141] Poese, I., Uhlig, S., Kaafar, M. A., Donnet, B., and Gueye, B. IP Geolocation Databases: Unreliable? *ACM SIGCOMM Computer Communication Review 41* (2011).

[142] Pujol, J., Erramilli, V., Siganos, G., Yang, X., Laoutaris, N., Chhabra, P., and Rodriguez, P. The Little Engine(s) That Could: Scaling Online Social Networks. In *ACM SIGCOMM* (2010).

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

[144] Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., and Shenker, S. Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure. In *HotNets* (2012).

[145] Rasti, A., Stutzbach, D., and Rejaie, R. On the Long-term Evolution of the Two-Tier Gnutella Overlay. In *IEEE Global Internet* (2006).

[146] Ratnasamy, S., Handley, M., Karp, R., and Shenker, S. Topologically Aware Overlay Construction and Server Selection. In *IEEE INFOCOM* (2002).

[147] Rekhter, Y., Li, T., and Hares, S. A Border Gateway Protocol 4 (BGP-4). RFC 4271, RFC Editor, Jan 2006.

[148] Rosenblum, M. Vmware's virtual platform$^{\mathrm{TM}}$. In *Hot Chips: A Symposium on High Performance Chips* (1999), pp. 185–196.

[149] University of Oregon Routeviews Project. `http://www.routeviews.org/`.

[150] Sahoo, J., Mohapatra, S., and Lath, R. Virtualization: A survey on concepts, taxonomy and associated security issues. In *Internation Conference on Computer and Network Technology* (2010).

[151] Sandvine Inc. Global Broadband Phenomena. Research Report `http://www.sandvine.com/news/global_broadband_trends.asp`, 2009-2011.

[152] Savage, S., Collins, A., and Hoffman, E. The End-to-End Effects of Internet Path Selection. In *ACM SIGCOMM* (1999).

[153] 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 Infastructure Systems and Architectures* (2009).

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

[155] Schulze, H., and Mochalski, K. Internet Study 2006-2009. `http://www.ipoque.com/resources/internet-studies`.

[156] Seetharaman, S., and Ammar, M. On the Interaction between Dynamic Routing in the Native and Overlay Layers. In *IEEE INFOCOM* (2006).

[157] Sekar, V., Ratnasamy, S., Reiter, M. K., Egi, N., and Shi, G. The middlebox manifesto: enabling innovation in middlebox deployment. In *HotNets* (2011).

[158] Shen, G., Wang, Y., Xiong, Y., Zhao, B. Y., and Zhang, Z.-L. HPTP: Relieving the Tension between ISPs and P2P. In *IPTPS* (2007).

[159] Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., and Sekar, V. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service. In *ACM SIGCOMM* (2012).

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

[161] Stavrou, A., Rubenstein, D., and Sahu, S. A lightweight, robust p2p system to handle flash crowds. In *International Conference on Network Protocols* (2002).

[162] Steiner, M., Gaglianello, B. G., Gurbani, V., Hilt, V., Roome, W. D., Scharf, M., and Voith, T. Network-aware service placement in a distributed cloud environment. *ACM SIGCOMM Computer Communication Review 42*, 4 (2012).

[163] Steinmetz, R., and Wehrle, K. *P2P Systems and Applications.* PWS Publishing Company, 2005.

[164] Streamingmedia Blog. Streamingmedia blog. `http://blog.streamingmedia.com/the_business_of_online_vi/2011/06`.

[165] Su, A., Choffnes, D., Kuzmanovic, A., and Bustamante, F. Drafting behind Akamai (travelocity-based detouring). In *ACM SIGCOMM* (2006).

[166] Su, A.-J., Choffnes, D. R., Kuzmanovic, A., and Bustamante, F. E. Drafting behind Akamai: Inferring Network Conditions based on CDN Redirections. *IEEE Trans. Communications 17*, 6 (2009).

[167] Tanenbaum, A. S. *Modern Operating Systems.* Prentice Hall Press, 2007.

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

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

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

[171] Virtual Computing Environment Consortium. `http://www.vce.com`.

[172] Vixie, P. What dns is not. *ACM Queue 7*, 10 (Nov 2009).

[173] Wang, J. A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review 29*, 5 (Oct 1999).

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

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

[176] Wilson, C., Ballani, H., Karagiannis, T., and Rowstron, A. Better Never then Late: Meeting Deadlines in Datacenter Networks. In *ACM SIGCOMM* (2011).

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

[178] Yang, X., and de Veciana, G. Service Capacity of Peer to Peer Networks. In *IEEE INFOCOM* (2004).

# A

# HTTP Methods and Status Codes

| Method | Description |
|---|---|
| GET | request the specified resource |
| HEAD | request only the metadata of the specified resource |
| POST | send data to the resource, e.g., input to a Web form |
| PUT | store or modify data under the specified ressource |
| DELETE | delete the specified resource |
| OPTIONS | get all possible methods available for the resource |
| TRACE | echoes back the request (for debugging purposes) |
| CONNECT | connect thrrough a proxy |
| PATCH | partially modify the resource |

Table A.1: HTTP methods defined by HTTP/1.1

| Code | Meaning | Examples |
|---|---|---|
| 1xx | Informational | 100 = request accepted but not complete |
| 2xx | Success | 200 = request to URI was successful |
| | | 204 = no content available under URI |
| 3xx | Redirection | 302 = content temporarily moved |
| | | 304 = cached content is still valid |
| 4xx | Client Error | 402 = access to URI is forbidden |
| | | 404 = content not found |
| 5xx | Server Error | 500 = internal server error |
| | | 503 = service unavailable, retry later |

Table A.2: HTTP status code groups defined by HTTP/1.1

# B
# CDI Measurement Object Sizes

| CDI1 | | CDI2 | |
|---|---|---|---|
| object# | size | object# | size |
| 01 | 38K | 01 | 36K |
| 02 | 66K | | |
| 03 | 154K | | |
| 04 | 217K | 02 | 254K |
| 05 | 385K | 03 | 471K |
| 06 | 510K | 04 | 599K |
| 07 | 905K | | |
| 08 | 2.48M | 05 | 3.4M |
| 09 | 6.16M | 06 | 4.5M |
| 10 | 17M | 07 | 8.6M |

Table B.1: CDI performance evaluation: Object sizes.