Improving Content Delivery with PaDIS

Content-delivery networks (CDNs) originate a large fraction of Internet traffic; yet, due to how CDNs often perform traffic optimization, users aren't always assigned to the best servers for end-user performance. To improve user assignment of CDNs, the authors propose and deploy the Provider-aided Distance Information System (PaDIS), which lets ISPs augment the CDN server by utilizing their unique knowledge about network conditions and user locations. Field tests show that using PaDIS can result in significant improvements in download time.

Ingmar Poese, Benjamin Frank, Bernhard Ager, and Georgios Smaragdakis
Deutsche Telekom Laboratories, Technische Universität Berlin

Steve Uhlig
Queen Mary, University of London

Anja Feldmann
Deutsche Telekom Laboratories, Technische Universität Berlin

The Internet is now a system in which users generate and share large amounts of content with other users via applications such as online social networks, video portals, one-click hosters (OCHs), Web services, wikis, blogs, and peer-to-peer (P2P) file-sharing applications. Multimedia content — including photos, music, and videos, as well as software downloads and updates — constitutes most Internet traffic. Recent studies report that most users access this information via HTTP, which accounts for more than 50 percent of Internet traffic (see www.sandvine.com/news/global_broadband_trends.asp).[1–3] HTTP traffic's prevalence is due in large part to increased streaming content (such as that on Youtube) and the popularity of OCH-offered content[4] (from sites such as rapidshare.com). Such popular content is hosted by the Internet's "hyper giants,"[1] which include large content providers such as Google and Yahoo as well as content-delivery networks (CDNs) such as Akamai and Limelight.[5] For simplicity's sake, we refer to all these various players simply as CDNs.

Although today's CDNs can optimize traffic flows, minimize costs, and bring content closer to end users, they come with some limitations. To overcome these limitations, we designed a Provider-aided Distance Information System (PaDIS), operated by an ISP, that helps ISPs improve user experiences by utilizing information about network bottlenecks and user locations. PaDIS fills a gap in the content-delivery landscape, in part because it takes into account ISP constraints and user performance. Although PaDIS is deployed at the moment as a platform for ISPs to improve content delivery, long-term, it's meant as a generic platform to help both ISPs and CDNs deliver content to users. So, we see PaDIS as an opportunity to think globally about content

 1089-7801/12/$31.00

delivery by including the network and its users in the picture.

CDN Limitations

To achieve high levels of performance and scalability, CDNs rely on distributed infrastructures. Some have even deployed servers deep inside ISPs in more than 5,000 locations throughout the Internet.[6] Others rely on placing numerous servers in data centers across strategic locations that provide good connectivity to major ISPs.[6,7]

Today, CDNs have full control over server-selection mechanisms, and can thus optimize traffic flows, minimize operational costs, and bring content closer to users. CDNs use two techniques to direct users to servers: DNS-based schemes or HTTP redirection. Because HTTP redirection yields a higher overhead and is more intrusive to the application architecture, DNS-based schemes are more widely used. Figure 1 presents the DNS-based approach's general architecture.

From user and ISP viewpoints, the redirection schemes that existing CDNs employ have three major limitations: network bottlenecks, user mislocations, and content-delivery costs.

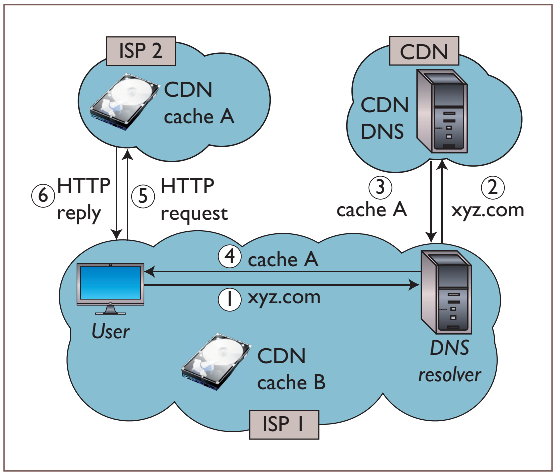

Figure 1. A DNS-based server-selection mechanism. Server selection occurs in the following steps: (1) The user issues a DNS request to its configured DNS resolver for the hostname associated with the content it wishes to retrieve. (2) The resolver asks the CDN's DNS server for an IP address that resolves to the requested hostname. (3) After the CDN has selected the servers based on this IP address, it returns them to the DNS resolver, which forwards the reply to the user. (5,6) With the IP address, the user retrieves the content via HTTP.

Network Bottlenecks

Despite the traffic-flow optimization that CDNs perform, assigning user requests to servers can still result in suboptimal content-delivery performance. This is a consequence of the limited information CDNs have regarding network conditions between users and their servers. Tracking the ever-changing conditions in networks (for example, through active measurements and user reports) incurs an overhead for the CDN without guaranteeing performance improvements for the user. Without sufficient information about network paths between CDN servers and users, any assignment the CDN performs can lead to either additional load on existing network bottlenecks or new bottlenecks.

User Mislocations

DNS requests that CDN DNS servers receive originate from users' DNS resolver, not from users themselves. The assignment is therefore based on the assumption that users are close to their DNS resolvers. Recent studies have shown that, in many cases, this assumption doesn't hold.[8,9] Consequently, the user is mislocated, and the server assignment isn't optimal. As a response to this issue, some have proposed DNS extensions to include the user's IP information.[10]

Content-Delivery Costs

Finally, CDNs strive to minimize the overall cost of delivering huge amounts of content to users. To this end, their assignment strategy is driven mainly by economic aspects. Although a CDN will always try to assign users in such a way that the server can deliver reasonable performance, these economic factors can result in users not being directed to the server that can deliver the best performance.

PaDIS

Today's content-delivery landscape is mostly unaware of information ISPs have about dynamic network conditions and user locations in the network. PaDIS lets an ISP influence server selection by extending its DNS infrastructure. To improve user performance, PaDIS leverages the server diversity of CDNs available through multiple locations (from which users can obtain content) as well as network information available only to the ISP.

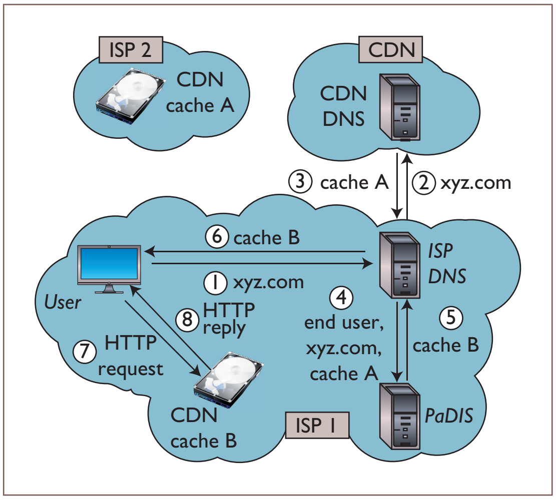

Figure 2. The incremental PaDIS deployment. Deploying PaDIS adds two steps to the DNS resolution process: the ISP's DNS infrastructure forwards the authoritative answer to the PaDIS server (4), which incorporates the answer into the server diversity, ranks all available servers based on the network information, and then returns the best server to the end host's DNS resolver (5), which in turn forwards the answer to the end host.

Currently deployed CDNs have significant server diversity.[11] We found that more than 70 percent of HTTP traffic in a large European ISP comes from at least three different network locations. We also found that only seven CDNs are responsible for more than 50 percent of HTTP traffic. Moreover, large-scale studies show that popular CDNs let users download content from any server.[12] To leverage available server diversity, PaDIS uses network information available only to the ISP to recommend (from a network perspective) the best server. Our evaluation of PaDIS provides evidence that the ISP's knowledge can improve the CDN's server-assignment process by avoiding the current limitations of DNS-based server selection.

PaDIS complements today's content-delivery landscape by adding an ISP optimization component. The PaDIS server's main tasks are to discover server diversity, maintain an up-to-date annotated map of the ISP network, and rank lists of available servers based on the network map.

Deployment

Deploying PaDIS inside an ISP is a two-step, incremental process. First, the ISP must modify its DNS infrastructure so that it can communicate with PaDIS. Because ISPs often operate multiple DNS resolvers, we must do this step-by-step to keep interruptions to a minimum. Furthermore, leveraging the DNS process requires users to employ the ISP DNS resolver, instead of a third-party resolver (such as OpenDNS or Google DNS). To this end, we verified that (in the large European ISP in which we chose to operate PaDIS) more than 97 percent of customers use the ISP-supplied DNS resolver.[11]

Second, the ISP must configure PaDIS to influence the server selection of either all CDNs or only some CDNs (for example, the most popular). Given that CDNs deliver different types of content (such as real-time content, websites, or bulk data), a PaDIS administrator can set diverse performance metrics for different CDNs.

Figure 2 presents a possible PaDIS deployment, co-located with the ISP DNS infrastructure. Three aspects are important for a successful PaDIS deployment: transparency, latency, and server aggregation. Obtaining cooperation from the user as well as from the CDN is challenging. Unless such cooperation is available, PaDIS deployment should be completely transparent to both. To achieve this, PaDIS intercepts and modifies a portion of the DNS messages involved in server selection, without requiring either the user or the CDN to change its current operation. Moreover, the communication delay between the ISP DNS resolver and PaDIS should be minimized to avoiding harming user performance. The best solution is to physically co-locate PaDIS with the DNS resolvers, which also lets PaDIS discover the largest possible server diversity in a CDN as unveiled from the authoritative DNS answers. The larger the user population utilizing the ISP DNS resolver, the higher the chance of discovering a large set of candidate servers — even in small time scales.

To scale PaDIS with growing ISP requirements, we designed our PaDIS prototype's architecture to keep up with the request rate of an ISP DNS resolver.[11] Moreover, the increase in the overall DNS resolution time is negligible.[11] Also, every PaDIS server has its own network feeds, letting it work independent of other PaDIS instances, and thus removing any overhead from synchronization.

Architecture

To achieve PaDIS's tasks, we propose an architecture that comprises a *diversity discovery* component, a *network monitoring* component, and a *query processing engine*. Figure 3 provides the PaDIS architecture overview.

Diversity discovery. Improving how users are assigned to servers requires knowledge about content servers' location diversity. However, CDNs don't provide a list of the content servers they operate and their locations. The diversity discovery component extracts CDN server diversity from DNS replies, which are observed by the ISP DNS resolver, and applies aggregation rules defined by the PaDIS administrator. The aggregation rules filter and build lists of IP addresses for servers that can satisfy user requests for content. With the help of these lists, the choices offered to the query processing engine for choosing a close-by content server enhance beyond just the individual DNS reply.

The diversity discovery component also acts as a protocol converter, providing an interface between the ISP DNS resolver and the other PaDIS components. It speaks the standard DNS protocol with the ISP DNS resolver, reducing the number of changes needed.

Network monitoring. ISPs have very detailed and up-to-date information about their own network infrastructure. Nonetheless, determining the overall network status involves several systems — each responsible for monitoring part of that status. The network monitoring component gathers information about the network's state from several sources and maintains an up-to-date view of the network. It also provides an interface for network-status queries.

The network monitoring component has three subcomponents. The *topology information* component gathers detailed information about the network topology as well as link utilization, router load, and topological changes. The *connectivity information* component uses routing information from within the network and other ISPs (in other words, through analyzing Border Gateway Protocol [BGP] messages) to enable a mapping from the topology to network paths. The *network map database* processes the data from the other two components to allow fast access to network information; it also abstracts the topology from a setup-specific setting into an annotated graph representation and pre-caches network paths along with their properties.

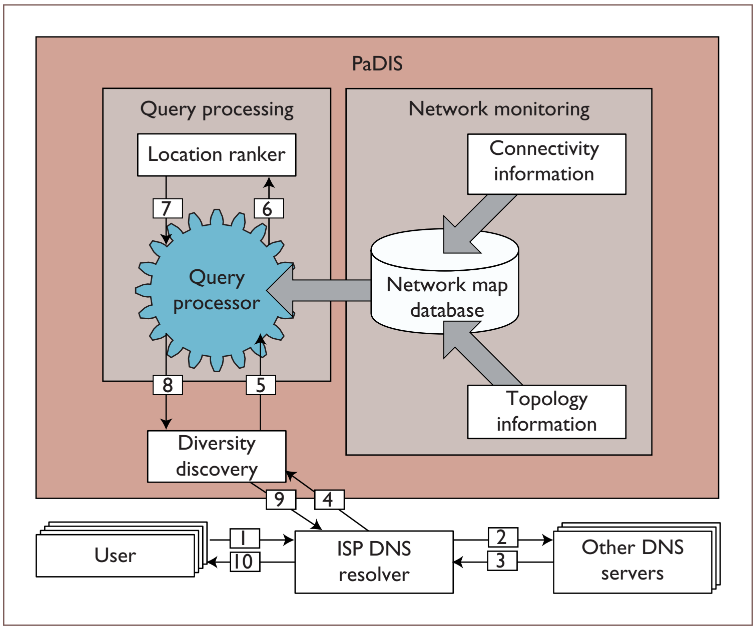

Figure 3. Processing of DNS requests from end users with PaDIS enabled. The PaDIS-enhanced DNS resolution processes this request as follows. (1) The user sends a DNS request for CDN-ized content optimized by PaDIS. (2) The ISP DNS resolver forwards the request to the appropriate, authoritative DNS servers if the answer isn't in the cache. (3) The ISP DNS resolver receives an answer to the query, either from the authoritative DNS server or from its own cache. (4) The ISP DNS resolver gives the DNS answer to PaDIS's diversity discovery component, which (5) enhances the DNS answer with additional sources that can satisfy the request from previous, similar DNS answers that have already been received. It then sends the complete list to the query processor, which (6) augments each source-destination pair from the received list with information from the network monitoring component and gives each pair, separately, to the location ranker. (7) The location ranker returns the preference value for the specific source-destination pair. Once all pairs have been given a preference, (8) the location ranker sorts the list of sources by the assigned preferences and sends it back to the diversity discovery component, which selects the top-ranked sources (servers), generates a DNS reply, and sends it back to the ISP DNS resolver. (10) The user receives the modified answer.

Having ISP-centric information ready for fast access in a database ensures a timely response if a decision based on the network state is required.

Query processing. The query processing engine combines information about server diversity and up-to-date network status to improve content delivery. It calculates the preference of a given set of candidate sources (servers) to reach a destination (user) while taking into account network conditions between the sources and the destination.

It comprises two subcomponents: the *query processor* and the *location ranker*.

The query processor augments each source-destination pair in the query with network conditions. Then it hands each pair, individually, to the location ranker to get its preference value. Finally, the query processor sorts the list of source-destination pairs by preference values and hands it back to the diversity discovery component. The location ranker receives exactly one source-destination pair along with the network conditions between them. To calculate a preference value, it applies a specific metric for optimization (such as delay, hop count, or link utilization).

PaDIS isn't only a platform for content-delivery optimization. Its modular architecture can be adapted to arbitrary protocols and network environments. For example, PaDIS currently gathers its information about server diversity from DNS replies; future versions of the diversity discovery component might obtain such information directly from the CDN or a third-party mapping system such as Application-Layer Traffic Optimization (ALTO; https://datatracker.ietf.org/wg/alto/charter/). Furthermore, PaDIS currently aims to optimize HTTP traffic from CDNs. However, the actual HTTP connection is indirectly influenced by the preceding server-selection process through DNS; thus, PaDIS leverages the decoupling of the server selection and content transfer, thus improving content delivery independent of the protocol used for content transfer. Moreover, PaDIS might enable collaboration between different parties involved in content delivery — namely, CDNs or other ISPs — to jointly improve their operation and the user experience. Note also that PaDIS doesn't unveil any sensitive information about ISP operation to other parties.

Experience with PaDIS

To quantify PaDIS's effects, we consider two different CDNs. Our choice is driven by the large traffic volume both CDNs carry in the major European ISP we consider to operate PaDIS. With help from anonymized, packet-level traces from this large European ISP, we selected a CDN that's responsible for more than 20 percent of that ISP's HTTP traffic; the CDN has highly distributed caches that typically deliver small- to average-sized files. The second CDN is an OCH — responsible for more than 15 percent of the ISP's HTTP traffic — that relies on a multi-homed data center to deliver large-sized files to users. After choosing CDNs to study, we deployed vantage points at residential locations with DSL connectivity in the ISP and performed extensive active measurements to evaluate PaDIS in the wild. The results presented here represent a much larger set of experiments; Ingmar Poese and colleagues describe those experiments in more detail.[11]

For the CDN, we utilized PaDIS's diversity discovery component to extract server diversity. The aggregated rule we used is based on the CDN's DNS redirection signature. We identified more than 3,500 unique servers operated by the large CDN spread over 124 different network locations. A large fraction of these caches are located within the studied ISP. Next, we randomly selected one cache from each location to run our experiments and confirmed that all the chosen caches serve the content we were downloading for measurement. Note that most CDNs serve all content from all their caches.[12] Because most files this CDN delivers have a small-to-average size, the download performance is dominated by the end-to-end delay;[13] thus, we used end-to-end delay as the metric to rank caches in PaDIS. During the experiment, we downloaded several files from all 124 chosen servers, including those the CDN chose and those PaDIS chose. Figure 4a shows the download time of an average-sized object, and the median download time across all 124 preselected caches. The reduction in download time was up to a factor of four when using PaDIS; when considering larger files (for example, more than 10 Mbytes), the improvement in download time was smaller because download performance is restricted by the network bandwidth on the bottleneck link, and the end-to-end delay becomes less significant.

Let's now look at the OCH. Using the same approach as with the other CDN, we determined that the OCH servers are located in a single data center. The OCH is well-connected to four large ISPs, including the large European ISP that operates PaDIS, through direct peering with their networks. A closer look at the OCH's server-selection strategy reveals that it assigns 60 percent of the large European ISP's user requests to servers using the direct peering links (that is, the traffic doesn't go through

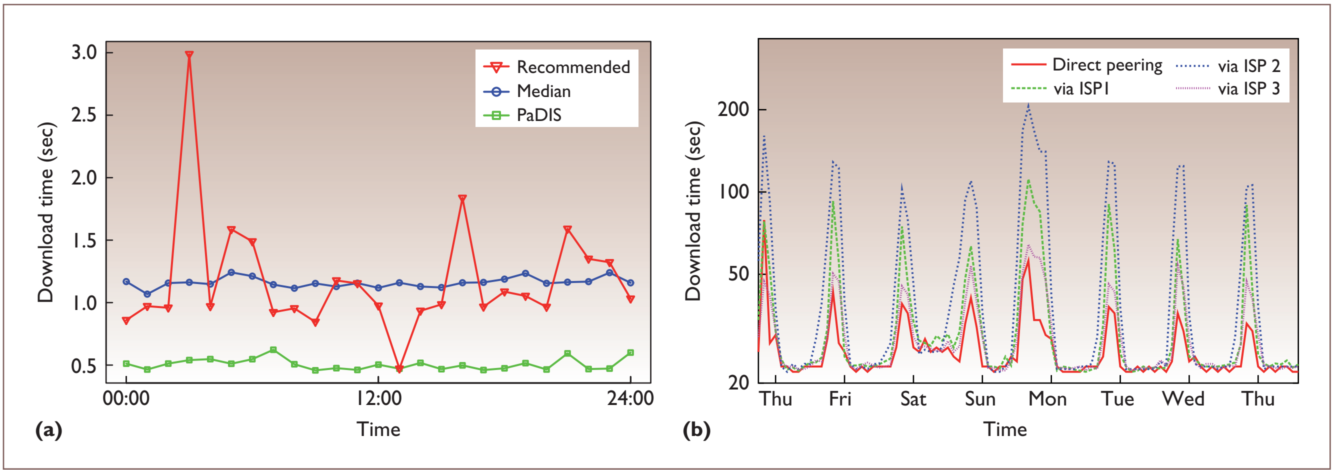

Figure 4. PaDIS evaluation with real ISP traces. We measured (a) the download time for a 510-Kbyte file distributed by a CDN. "Recommended" refers to the download time when PaDIS uses the CDN's cache recommendation, "median" to the download time of all available CDN caches PaDIS has discovered; and "PaDIS" as the download time when using PaDIS. We also measured (b) the download time for a 50-Mbyte file from an OCH; PaDIS utilizes direct peering with the OCH, improving the download time by a factor of four during the peak hour.

any other ISP), regardless of the time of day or the day of the week. The OCH randomly assigns remaining requests to servers behind peerings with the other three ISPs. Knowing this, we set PaDIS's ranking function such that the OCH servers behind direct peering with the large European ISP are ranked higher than the others. This way, PaDIS always prefers direct peering, thus removing any additional penalties that come from going through a third network. We performed an experiment where we repeatedly downloaded content using the different OCH peerings. In Figure 4b, we plot the download time of a 50-Mbyte file over a one-week period. Improvement in download time was up to a factor of four during the peak hour, when network resources were scarce.

Expected gains from deploying PaDIS depend on several aspects, such as the number of locations from which given content is available, the properties of the paths available between the servers and end users, and the type and volume of the requested content. CDNs continue to expand their infrastructures to keep up with increasing traffic demands. Thus, deploying PaDIS will be more beneficial in the future, because it can take advantage of the increased server diversity CDNs expose.

PaDIS leverages the decoupling of server selection and content transfer; thus, it isn't restricted to DNS-based server selection and "CDN-ized" traffic, but can be utilized by any protocol. As part of our future work, we plan to investigate how we can utilize PaDIS to improve the content-delivery performance of applications that don't depend on DNS, as well as evaluate PaDIS in other ISPs.

References

1. C. Labovitz et al., "Internet Inter-Domain Traffic," *Proc. ACM SIGCOMM 2010 Conf.*, ACM Press, 2010, pp. 75–86.
2. G. Maier et al., "On Dominant Characteristics of Residential Broadband Internet Traffic," *Proc. 9th ACM SIGCOMM Internet Measurement Conf.*, ACM Press, 2009, pp. 90–102.
3. "Cisco Visual Networking Index: Forecast and Methodology, 2010–2015," white paper, Cisco; June 2011; www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf.
4. D. Antoniades, E.P. Markatos, and C. Dovrolis, "One-Click Hosting Services: A File-Sharing Hideout," *Proc. 9th ACM SIGCOMM Internet Measurement Conf.*, ACM Press, 2009, pp. 223–234.
5. C. Huang et al., "Measuring and Evaluating Large-Scale CDNs," *Proc. 8th ACM SIGCOMM Internet Measurement Conf.*, ACM Press, 2008, pp. 15–29.
6. T. Leighton, "Improving Performance on the Internet," *Comm. ACM*, vol. 52, no. 2, 2009, pp. 44–51.
7. R. Krishnan et al., "Moving Beyond End-to-End Path Information to Optimize CDN Performance," *Proc. 9th ACM SIGCOMM Internet Measurement Conf.*, ACM Press, 2009, pp. 190–201.

8. Z. Mao et al., “A Precise and Efficient Evaluation of the Proximity between Web Clients and Their Local DNS Servers,” *Proc. Usenix Ann. Technical Conf,* Usenix Assoc., 2002, pp. 229–242.
9. B. Ager et al., “Comparing DNS Resolvers in the Wild,” *Proc. 10th ACM SIGCOMM Internet Measurement Conf.,* 2010, pp. 15–21.
10. C. Contavalli et al., “Client Subnet in DNS Requests,” IETF Internet draft, work in progress, 2011.
11. I. Poese et al., “Improving Content Delivery Using Provider-aided Distance Information,” *Proc. 10th ACM SIGCOMM Internet Measurement Conf.*, ACM Press, 2010, pp. 22–34.
12. S. Triukose, Z. Al-Qudah, and M. Rabinovich, “Content Delivery Networks: Protection or Threat?” *Proc. European Symp. Research in Computer Security* (ESORICS 09), Springer, 2009, pp. 371–389.
13. J. Padhye et al., “Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation,” *IEEE/ACM Trans. Networking*, vol. 8, no. 2, 2000, pp. 133–145.

Ingmar Poese is a PhD candidate at Deutsche Telekom Laboratories, Technische Universität Berlin. His research interests include network architectures, network measurements, and content distribution. Poese has a Diplom in computer science from Technische Universität Berlin. Contact him at ingmar@net.t-labs.tu-berlin.de.

Benjamin Frank is a PhD candidate at Deutsche Telekom Laboratories, Technische Universität Berlin. His research interests include the measurement, analysis, and optimization of content distribution systems and architectures as well as cloud systems. Frank has an MS in computer science from the Technical University of Munich. Contact him at bfrank@net.t-labs.tu-berlin.de.

Bernhard Ager is a research scientist at Deutsche Telekom Laboratories, Technische Universität Berlin. His research interests include passive and active Internet measurement, behavioral characterization, and content distribution. Ager has a PhD in computer science from Technische Universität Berlin. Contact him at bernhard@net.t-labs.tu-berlin.de.

Georgios Smaragdakis is a senior research scientist at Deutsche Telekom Laboratories, Technische Universität Berlin. His research interests include the measurement, performance evaluation, and optimization of content distribution systems and overlay networks. Smaragdakis has a PhD in computer science from Boston University. Contact him at georgios@net.t-labs.tu-berlin.de.

Steve Uhlig is a professor of networks and head of the networks research group at Queen Mary, University of London. His research interests focus on the large-scale behavior of the Internet, especially through network measurements, and include software-defined networking, content delivery, and network infrastructure virtualization. Uhlig has a PhD in applied sciences from the University of Louvain, Louvain-la-neuve, Belgium. Contact him at steve@eecs.qmul.ac.uk.

Anja Feldmann is a professor at Deutsche Telekom Laboratories, Technische Universität Berlin, where she heads the intelligent networks research group and is dean of the Department of Electrical Engineering and Computer Science. Her research interests include Internet measurements, programmable networks, and network architecture. Feldmann has a PhD in computer science from Carnegie Mellon University. Contact her at anja@net.t-labs.tu-berlin.de.